

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Virtualizace aplikací v architektuře C/S

Štěpán Hlaváč

© 2016 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Štěpán Hlaváč

Informatika

Název práce

Virtualizace aplikací v architektuře C/S

Název anglicky

Architecture C / S – Application Virtualization

Cíle práce

Hlavním cílem diplomové práce je identifikace aktuálních technologií v oblasti virtualizace aplikací a implementace zvolené technologie.

Dílní cíle:

- stanovení požadavků na systém
- analýza a následné porovnání technologií a přístupů k virtualizaci
- návrh a implementace vybrané technologie v reálném prostředí
- formulace závěrů, zhodnocení silných a slabých stránek

Metodika

Diplomová práce bude založena na studiu a analýze odborných literárních zdrojů.

V teoretické části práce budou porovnávány přístupy jednotlivých technologií k virtualizaci aplikací.

V praktické části bude proveden návrh a následná implementace zvolené technologie. Implementované řešení bude posléze testováno.

V závěru diplomové práce budou syntetizovány výsledky a formulována doporučení.

Doporučený rozsah práce

50-60 stran

Klíčová slova

virtualizace, App-V, Hyper-V, VDI, Remote Desktop, TCP/IC, server, PowerShell

Doporučené zdroje informací

- AL], Thomas Olzak .. [et]. a Nelson RUEST. Microsoft virtualization: master Microsoft Server, dektop, application, and presentation virtualization. Vyd. 1. Burlington, MA: Syngress, 2010, 408 s. ISBN 15-974-9431-3.
- KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS: master Microsoft Server, dektop, application, and presentation virtualization. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.
- KELBLEY, John a Libor DOSTÁLEK. Microsoft Windows Server 2008 R2 Hyper-V: podrobný průvodce administrátora. 5., aktualiz. vyd. Brno: Computer Press, 2011, 392 s. ISBN 978-80-251-3286-9.
- RUEST, Danielle a Nelson RUEST. Virtualizace: podrobný průvodce. Vyd. 1. Brno: Computer Press, 2010, 408 s. ISBN 978-80-251-2676-9
- WILSON, Ed a Libor DOSTÁLEK. PowerShell: průvodce skriptováním. 1. vydání. Brno: Computer Press, 2015, 263 stran. ISBN 8025143864

Předběžný termín obhajoby

2015/16 LS – PEF

Vedoucí práce

doc. Ing. Zdeněk Havlíček, CSc.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 28. 10. 2015

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 11. 11. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 30. 03. 2016

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Virtualizace aplikací v architektuře C/S" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.3.2016

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Zdeňku Havlíčkovi, CSc. za odborné vedení a poskytnuté rady v průběhu psaní této práce. Dále bych chtěl poděkovat své rodině a blízkým za podporu během studia.

Virtualizace aplikací v architektuře C/S

Souhrn

Tato práce pojednává o možnostech využití virtualizace aplikací v prostředí Windows. V teoretické části je popsán princip fungování aplikační virtualizace. Dále jsou specifikovány jednotlivé charakteristické vlastnosti a důvody pro přechod k virtualizaci aplikací. Je poskytnut základní přehled a porovnání konkrétních způsobů virtualizace. Na základě Saatyho metody je vybráno vhodné řešení pro praktickou část práce, v níž je detailně popsán postup implementace a používání zvolené technologie. Následuje zhodnocení fungování daného řešení na základě praktických zkušeností. V závěru práce jsou formulována obecná i specifická doporučení pro využívání aplikační virtualizace.

Klíčová slova: virtualizace, App-V, Hyper-V, VDI, Remote Desktop, TCP/IP, server, klient, PowerShell

Architecture C/S – Application Virtualization

Summary

The diploma thesis deals with the possibilities of using application virtualization in the Windows environment. The theoretical part describes the operating principle of application virtualization. Furthermore, there are specified the individual characteristics and the reasons for the move to application virtualization. The author provides the basic overview and comparison of the specific ways of virtualization. The appropriate solution of the practical part was based on the Saaty method. In the practical part, the author describes in detail how to implement and use the chosen technology. Based on practical experience, he assesses functioning of the solution. The final part of the thesis includes both general and specific recommendations for the use of application virtualization.

Keywords: virtualization, App-V, Hyper-V, VDI, Remote Desktop, TCP/IP, server, client, PowerShell

Obsah

1 Úvod	10
2 Cíl práce a metodika	11
2.1 Cíl práce.....	11
2.2 Metodika.....	11
3 Teoretická východiska	12
3.1 Charakteristika aplikační virtualizace	13
3.1.1 Proces vytváření balíčků aplikací	14
3.1.2 Virtualizace prezenční vrstvy	14
3.1.3 Virtualizační vrstva	15
3.1.4 Nezávislost na verzi Windows.....	16
3.1.5 Využití koncového agenta	16
3.1.6 Využití lokální mezipaměti	17
3.1.7 Využití Active Directory	17
3.1.8 Streaming aplikací.....	17
3.2 Přehled technologií.....	19
3.3 Microsoft Application Virtualization (App-V).....	22
3.3.1 Komponenty APP-V	23
3.3.2 App-V módy	25
3.4 ThinApp.....	25
3.4.1 Vytváření balíčku.....	26
3.4.2 Nasazení balíčku	26
3.5 Výběr vhodné technologie.....	28
4 Vlastní řešení.....	29
4.1 Sequencer.....	30
4.1.1 Předpoklady a systémové požadavky.....	30
4.1.2 Instalace	31
4.1.3 Vytváření aplikačních balíčků	32
4.2 Serverová část	34
4.2.1 Předpoklady a systémové požadavky.....	35
4.2.2 Instalace a konfigurace	35
4.2.3 Správa a distribuce virtuálních aplikací	38
4.3 Klientská část	40
4.3.1 Předpoklady a systémové požadavky.....	40
4.3.2 Instalace a konfigurace	41
4.3.3 Práce s aplikací	43

5 Zhodnocení výsledků a doporučení.....	46
6 Závěr	48
7 Seznam použitých zdrojů	50

Seznam obrázků

Obrázek 1 - Model virtualizace, zdroj: [3]	12
Obrázek 2 – Schéma streamovacího mechanismu, zdroj: [6]	19
Obrázek 3 - Schéma virtuálních aplikací, zdroj: [12]	22
Obrázek 4 - Schéma distribuce ThinApp, zdroj: [13]	27
Obrázek 5 - Úvodní instalační okno, zdroj: vlastní pořízení	31
Obrázek 6 - Příprava PC, zdroj: vlastní pořízení	32
Obrázek 7 - Specifikace cílového OS, zdroj: vlastní pořízení	34
Obrázek 8 - Výběr komponent App-V Serveru, zdroj: vlastní pořízení	36
Obrázek 9 - Nastavení Serveru App-V, zdroj: vlastní pořízení	37
Obrázek 10 - První spuštění administrátorské konzole, zdroj: vlastní pořízení	38
Obrázek 11 - Správa aplikačních balíčků, zdroj: vlastní pořízení	39
Obrázek 12 - Konfigurace aplikačního balíčku, zdroj: vlastní pořízení.....	40
Obrázek 13 - První spuštění klienta, zdroj: vlastní pořízení.....	41
Obrázek 14 - Přidání publikačního serveru, zdroj: vlastní pořízení.....	42
Obrázek 15 - Úvodní obrazovka klienta, zdroj: vlastní pořízení	44
Obrázek 16 - Virtuální aplikace v režimu online, zdroj: vlastní pořízení	45

Seznam tabulek

Tabulka 1 – Výhody a nedostatky virtualizace prezenční vrstvy, zdroj: [1]	14
Tabulka 2 - Porovnání tradičního a virtualizovaného prostředí, zdroj: [8]	15
Tabulka 3 - Přehled virtualizačních technologií, zdroj: [1].....	20
Tabulka 4 - Vlastnosti virtualizačních řešení, zdroj: [1]	21
Tabulka 5 - Porovnání technologií Saatyho metodou, zdroj: vlastní pořízení	28

1 Úvod

V dnešní digitální době si lze jen obtížně představit život bez informačních technologií. Počítače nahradily zastaralé kartotéky, sofistikované aplikace zase kalkulačku. Integrace informačních technologií se dotkla všech pracovních odvětví. Na základě toho dnes vznikají specializované softwary určené pro nejrůznější předměty činností v oblasti marketingu, financí, vývoje, lidských zdrojů a mnohé další.

U větších společností tak vzniká široká poptávka po programovém vybavení. V rámci jednotlivých oddělení se vytvářejí různé množiny aplikačního softwaru. Kromě těchto specifických množin existuje i základní sada programového vybavení, která je využívána napříč celou firmou. Zejména se jedná o poštovní klienty, textové editory a podobně. Poslední skupinou jsou speciální aplikace určené pro konkrétní jednotlivce, typicky pro vedoucí pracovníky. Zároveň aplikace procházejí takzvaným životním cyklem, který nezbytně zahrnuje optimalizace v podobě aktualizací. Díky aktualizacím existují stejné aplikace v různých verzích. Cílem správce programového vybavení je tyto verze sjednocovat, aby bylo zachováno konzistentní chování koncových stanic. Samotnému nasazení aktualizací by mělo předcházet testování kompatibility s ostatními aplikacemi.

Výsledkem výše zmíněného je potřeba efektivně spravovat a distribuovat aplikace využívané v rámci podniku. Existují různé přístupy k distribuci a správě aplikací. Jednou z možností je využití virtualizace aplikací. Přináší možnost centralizované správy a distribuce aplikací. Dále řeší problém konfliktů mezi jednotlivými aplikacemi, jelikož jsou aplikace izolované od operačního systému a od sebe navzájem.

V rámci této práce bude provedena analýza jednotlivých přístupů k virtualizaci. Posléze budou konkrétní způsoby řešení porovnány. Následuje vyhodnocení a zvolení nejvhodnější technologie, splňující specifické požadavky daného prostředí. Vybrané řešení bude implementováno v reálném prostředí. V závěru práce bude provedeno zhodnocení nasazeného řešení a budou formulována doporučení.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem diplomové práce je identifikace aktuálních technologií v oblasti virtualizace aplikací a implementace zvolené technologie v reálném prostředí. Aplikační virtualizace bude realizována na platformě Windows. V konečném důsledku by zvolená technologie představovala centralizované řešení pro správu aplikačního portfolia.

Dílčí cíle:

- Zdůraznit teoretické principy virtualizace
- Analyzovat a porovnat virtualizační technologie
- Implementovat vybrané řešení virtualizace aplikací
- Formulovat obecné i specifické závěry

2.2 Metodika

Diplomová práce bude založena na studiu a analýze odborných literárních zdrojů. V teoretické části práce budou porovnávány přístupy jednotlivých technologií k virtualizaci aplikací. V praktické části bude proveden návrh a následná implementace zvolené technologie. Implementované řešení bude posléze testováno. V závěru diplomové práce budou syntetizovány výsledky a formulována doporučení.

3 Teoretická východiska

Počátky virtualizace sahají do konce 60. let minulého století. Průkopníkem v oblasti virtualizace byla firma IBM. Za tuto dlouhou dobu prošla virtualizace transformací, která v současné podobě představuje technologii určenou nejen pro IT specialisty, ale i pro běžné uživatele výpočetní techniky. Dále se stala základem pro utváření firemní IT struktury. Existují různé formy virtualizace. Virtualizace může probíhat například na úrovni serveru. Virtualizace serverových operačních systémů přináší nezávislost na hardwaru, což umožňuje migrovat virtuální počítač na jiný hardwarový prostředek během pracovního dne. Virtualizovány jsou i koncové stanice, kdy uživatel pracuje na svém zařízení ve virtuálním prostředí. Virtualizovat lze i samotné aplikace. Kompletní model virtualizace je znázorněn na následujícím obrázku. [3]



Obrázek 1 - Model virtualizace, zdroj: [3]

Serverová virtualizace (SerV) umožňuje virtualizovat libovolný operační systém založený na platformě x86 nebo x64. Přičemž existuje softwarová a hardwarová virtualizace. U softwarové virtualizace běží vizualizovaný operační systém nad softwarovou vizualizační platformou na skutečném operačním systému. Hardwarová virtualizace využívá softwarovou platformu, pro virtualizaci operačního systému, přímo

nad hardwarem bez použití existujícího operačního systému. Virtualizace úložišť (StoreV) slouží k složení fyzického úložiště z více zařízení tak, aby navenek vypadalo jako jeden. Virtualizace sítí (NetV) umožňuje ovládat šířku pásma jejím rozdělením na nezávislé kanály. Ty lze pak přiřadit jednotlivým zdrojům. Typickým příkladem může být technologie VLAN. Správa virtualizace (ManageV) slouží k správě datového centra. Jedná se o fyzické i virtuální prostředky, které jsou prezentovány jako jediná sjednocená infrastruktura pro poskytování služeb. Virtualizace desktopů (DeskV) představuje virtuální počítače pro distribuci desktopových systémů. Virtualizace prezenční vrstvy (PresentV), dříve označovaná jako terminálové služby, nabízí uživatelům prezenční vrstvu z centrálního umístění. Tato vrstva je postupně nahrazována jinými technologiemi jako je například AppV. Virtualizace aplikací (AppV) využívá principy serverové virtualizace s tím rozdílem, že pouze odděluje provozní aplikace od operačního systému. [3, 4]

Na základě výše uvedeného rozdělení lze konstatovat, že virtualizace je široce zaměřená problematika. Z tohoto důvodu si tato práce neklade za cíl pokrýt celou šíři, nýbrž se věnuje pouze problematice virtualizace aplikací. Speciálně je kladen důraz na technologie typu klient-server. V následujících kapitolách bude podrobně popsána problematika virtualizace aplikací. Budou specifikovány konkrétní technologie. Dále bude provedeno porovnání a zhodnocení konkrétních řešení.

3.1 Charakteristika aplikační virtualizace

Jedním z hlavních důvodů pro využití virtualizace je samotná struktura systému Windows. Systém představuje prostředek pro využívání aplikací. Základní funkce jsou integrovány do jádra systému. Aplikace využívající tyto funkce pouze volají součásti systému. Nicméně tyto funkce (knihovny DLL) se v průběhu času mění. To může zapříčinit nefunkčnost volané funkce. Z tohoto důvodu jsou tyto knihovny instalovány společně s aplikací. Funkčnost instalované aplikace je tím zajištěna. Změnou knihovny je však ohrožena funkčnost již instalovaných aplikací. Nejen tento problém řeší virtualizace aplikací.

Jak již bylo zmíněno, virtualizace aplikací odděluje aplikace nebo služby od operačního systému. Toho je docíleno pomocí virtualizační vrstvy. Tento přístup odpovídá výše zmíněné softwarové serverové virtualizaci. Z toho vyplývá, že pro fungování je vyžadován operační systém. Tem je ovšem v případě virtuálních aplikací chráněn před

změnami, které standardní aplikace během instalace mohou provádět. Během vytváření virtuální aplikace je zaznamenávám stav spuštěné aplikace a všeho ostatního co aplikace využívá. Posléze je aplikace přenesena na koncovou stanici bez nutnosti instalace. [1]

3.1.1 Proces vytváření balíčků aplikací

Tento proces se sestává z pěti nezbytných kroků. První krok se zaměřuje na správu požadavků. Předmětem je analýza současných aplikací, která zajistí zavedení pouze nezbytných aplikací do systému. Druhý krok je věnován přípravám na integraci dané aplikace. Především jsou zkoumány požadavky aplikace, kooperace s ostatními aplikacemi a samotný instalační proces. V tomto kroku je vytvořen prvotní balíček. Po provedení aplikační analýzy následuje třetí krok, který testuje počáteční balíček. Ověřuje se celková funkčnost a soulad s bezpečnostními předpisy. Ve čtvrtém kroku je kontrolována kvalita balíčku. Testuje se přidávání, odebrání, a zda nedochází ke konfliktu balíčku na koncové stanici. Posledním krokem je přidání balíčku do správy verzí. V tomto kroku je již připraven k nasazení na produkční prostředí. [1]

3.1.2 Virtualizace prezenční vrstvy

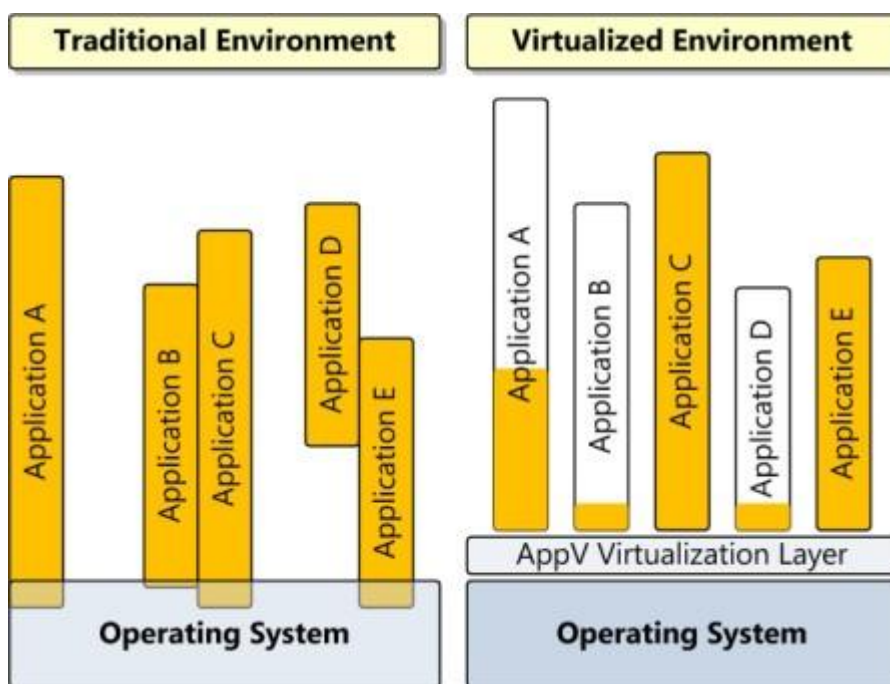
Princip virtualizace prezenční vrstvy spočívá v serverových instalacích, které jsou sdíleny mezi uživateli. Uživatelé k těmto aplikacím přistupují pomocí protokolu RDP. Tento přístup přináší značné výhody, ale zároveň i některé nedostatky.[1]

Výhody	Nedostatky
Minimalizace množství možných konfliktů	Značný výpočetní výkon na straně serveru
Značné snížení počtu aktualizacích umístění	Obtížnější konfigurace serveru
Integrované virtualizační funkce ve Windows Server	Nutná konektivita mezi serverem a klientem
Snížení časové náročnosti na nasazení a správu	

Tabulka 1 – Výhody a nedostatky virtualizace prezenční vrstvy, zdroj: [1]

3.1.3 Virtualizační vrstva

Princip virtualizační vrstvy spočívá ve vytvoření ochranné vrstvy nad operačním systémem, která systém chrání před změnami, které jsou s instalací aplikací spojeny. Hlavní princip spočívá v zachycení stavu spuštěné aplikace. Tento stav je posléze distribuován na cílovou stanici, kde nedochází ke změnám v operačním systému jako při klasické instalaci. Virtualizační vrstva má za úkol překládat stav aplikace na instrukce operačního systému. Tím je zajištěna komunikace mezi ostatními aplikacemi. Dále pak je zajištěno, že jádro operačního systému není ovlivněno a nevytváří se konflikty mezi ostatními aplikacemi. To je základní rozdíl oproti klasické instalaci, kdy aplikace pronikají do operačního systému a mění jeho nastavení. Graficky je tato situace znázorněna na následujícím obrázku. [8]



Tabulka 2 - Porovnání tradičního a virtualizovaného prostředí, zdroj: [8]

Vzhledem k izolaci aplikací od operačního systému a aplikací mezi sebou navzájem, přináší toto řešení další užitečnou vlastnost. Lze totiž provozovat více verzí jedné aplikace na jednom systému zároveň. Díky této vlastnosti lze spouštět jednotlivé verze aplikací, které při klasické koncepci spolu nemohou koexistovat (IE, Access, ...).

Virtualizaci aplikací lze provozovat jako samostatný celek, který pracuje na jednotlivých systémech. Toto řešení však není vhodné pro podnikové řešení. Vhodnější je centrální distribuce aplikací.

V zásadě existují dva způsoby:

- Tradiční (push, neboli vynucená)
- Streamovací (pull, neboli vyžádaná)

Při tradičním způsobu distribuce je aplikace doručena na koncové zařízení celá a plně nakonfigurována. V případě streamování jsou odeslány nejprve data bezpodmínečně nutná ke spuštění aplikace. Zbytek dat bude zaslán při práci s danou aplikací. Streamování dále umožňuje doručovat pouze aplikace, které chce daný uživatel právě využívat, nikoli všechny najednou. [1]

3.1.4 Nezávislost na verzi Windows

Virtualizace aplikací na platformě Windows podporuje vícero verzí tohoto systému. To umožňuje nasazovat již vytvořený aplikační balíček na různé operační systémy, bez nutnosti vytvářet balíček znovu. Tuto vlastnost zajišťuje engine virtualizace, který zprostředkovává komunikaci mezi operačním systémem a samotnou aplikací. K tomu využívá překlad instrukcí do požadovaného formátu dle konkrétního operačního systému. Jedinou nutností je aktualizace virtualizačního enginu na požadovanou verzi operačního systému. Tato vlastnost značně usnadňuje případný přechod na nový operační systém.[1]

3.1.5 Využití koncového agenta

Koncovou stanicí, na které běží virtualizace aplikací, lze provozovat s klientem, nebo bez něj. V prvním případě je před samotným spuštěním virtuální aplikace nutné mít na koncové stanici nainstalovaného klienta. Ten komunikuje s centrálním serverem. Agent může být nasazen standardním distribučním systémem, součástí bitové kopie systému, nebo nasazen pomocí streamovací služby. Druhá varianta explicitně neobsahuje agenta, nicméně je integrován přímo ve virtualizované aplikaci. Díky této vlastnosti odpadá starost s nasazením agentů na koncové stanice před samotným spuštěním virtualizace. Na druhou stranu je třeba dbát větší zřetel na bezpečnost. Virtuální aplikace bez klientů jsou přenosné. Z tohoto důvodu je na místě věnovat pozornost licenčním a zabezpečujícím mechanismům.[1]

3.1.6 Využití lokální mezipaměti

Virtualizace pro správné fungování využívá mezipaměť koncové stanice. Ta představuje dočasný prostor pro danou aplikaci. Přístupnost tohoto úložiště lze nadefinovat. Může být nestálé, což způsobí uvolnění mezipaměti po odhlášení uživatele. Dále lze mezipaměť nakonfigurovat jako dočasnou. V tomto případě lze specifikovat časový úsek, po který lze k mezipaměti přistupovat. Po uplynutí časového intervalu dojde k uvolnění mezipaměti a aplikace již nebude přístupná. Poslední možností je permanentní přístup.

Modifikováním přístupu k mezipaměti lze získat kontrolu nad využíváním virtualizovaných aplikací. V případě potřeby lze snadno změnit konfiguraci. Změna se provede na centrálním úložišti. Při následném spuštění aplikace automaticky ověří, zda nedošlo k nějaké změně. Při vyhodnocení změny se provede aktualizace, která modifikuje lokální mezipaměť. [1]

3.1.7 Využití Active Directory

Pro efektivní správu aplikací je využívána služba Active Directory (AD). U konkrétních aplikací lze specifikovat AD skupiny, které budou mít možnost dané aplikace využívat. Lze tak jednoduše přiřazovat specifické aplikace jednotlivým oddělením v rámci firmy. Pokud je uživatel členem AD skupiny, která má přiřazenou danou aplikaci, pak se mu zobrazí daná aplikace na pracovní ploše. Aplikace lze přiřazovat globálně. Tyto aplikace využívají všichni uživatelé systému. Dále aplikace přiřazené v rámci role, která utváří určitou skupinu uživatelů (oddělení). Poslední způsob je přiřazování jednotlivým uživatelům.[1]

3.1.8 Streaming aplikací

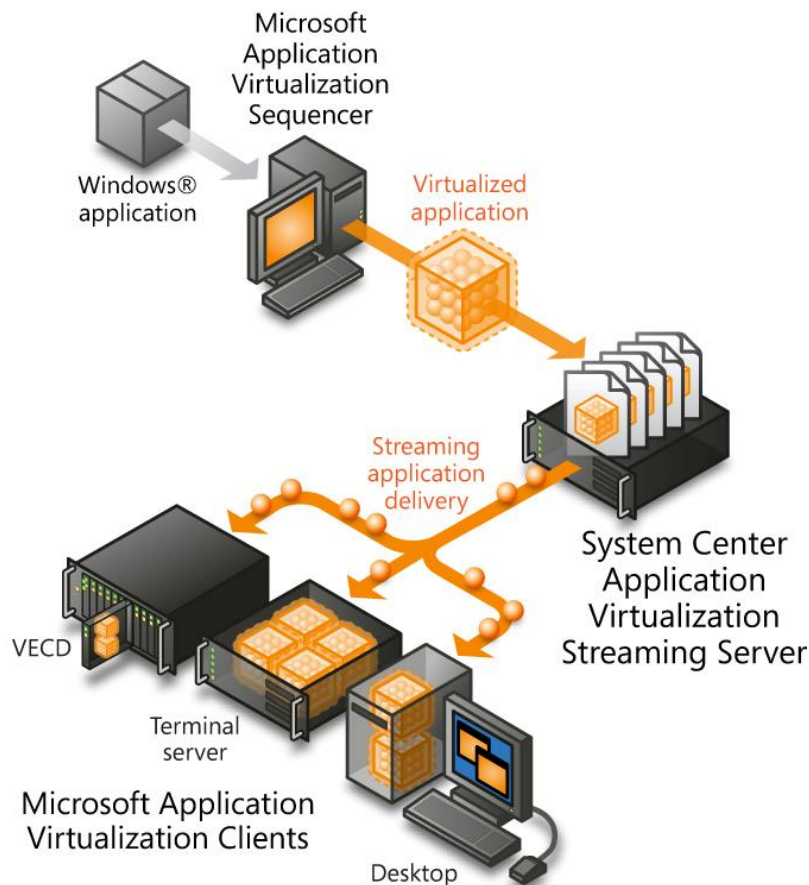
Streaming představuje způsob doručování aplikací na koncové stanice. Díky struktuře vytvořených aplikačních balíčků lze tyto kusy kódu rozdělit na segmenty. Tyto segmenty lze rozdělit do následujících kategorií a posléze streamovat na koncová zařízení:

- Spouštěcí – Blok kódu nutný ke spuštění aplikace. Jedná se o zlomek velikosti aplikace, který po stažení umožňuje s aplikací začít pracovat.

- Prediktivní – Jedná se o kusy kódu, které jsou na základě analýzy streamovacího serveru označeny za často využívané. Posléze je těmto blokům nastaveno automatické streamování na koncové stanice. Tím je docíleno rychlejší odezvy v případě požadavku na funkci, která ještě nebyla požadována a která je obsažena v prediktivním bloku. Prediktivní bloky se stahují zpravidla ihned po spouštěcích blocích.
- Na vyžádání – Jedná se o kusy kódu, které jsou doručovány na základě požadavku od uživatele. Jsou to funkce, které jsou využívány jen minimálně.

Díky výše zmíněnému lze začít využívat aplikaci o velikosti několika stovek megabajtů již po stažení jednotek, maximálně desítek megabajtů. Po stažení celého aplikačního balíčku je aplikace dostupná z lokální mezipaměti v režimu offline.

Tato technologie dále implementuje správu licencí. Ta spočívá v evidenci právě využívaných aplikací. Licence, které nejsou právě využívány lze rychle a opakovaně použít. Tím lze snížit náklady na licence. Streamovací proces nevyžaduje složitou infrastrukturu. V základu stačí pouze jeden streamovací server. Na následujícím obrázku je znázorněno schéma streamovacího procesu. [1]



Obrázek 2 – Schéma streamovacího mechanismu, zdroj: [6]

Na schématu je znázorněna virtualizovaná aplikace, která byla vytvořena za pomoci aplikace Sequencer. Tato aplikace posléze vstupuje do streamovacího serveru. Ten má za úkol doručovat jednotlivé bloky kódu na patřičná koncová zařízení.

3.2 Přehled technologií

S rostoucím rozmachem aplikační virtualizace roste i počet nabízených řešení. Mezi hlavní dodavatele virtualizace patří společnosti Citrix, Microsoft a VMware. Existují však i další méně známe firmy. Jednotlivé přístupy k virtualizaci se mohou v určitých aspektech lišit, ale základní principy jsou zachovány. V následující tabulce jsou uvedeni přední výrobci a jejich stručná charakteristika.

Výrobce	Popis
Symantec Endpoint Virtualization Suite	Ukládá aplikace a data do virtuálních softwarových balíčků. Podporuje okamžitou aktivaci a deaktivaci aplikací. Zamezuje konflikty mezi aplikacemi. Umožňuje streamování aplikací na požádání.
Citrix XenApp	Představuje kombinaci prezenční vrstvy a virtualizace aplikací. Podporuje streamování aplikací na koncová zařízení. Existuje v několika verzích.
Microsoft App-V	Produkt umožňuje spravovat virtuální aplikace a streamovat je na koncová zařízení (osobní PC, virtuální PC,...).
VMware ThinApp	Aplikace, dříve označovaná jako Thinstall Application Virtualization Suite, umožňuje vytvářet balíčky aplikací, které lze spustit bez instalace.
InstallFree Bridge	Toto řešení nepotřebuje ke svému používání klienta. Zajišťuje ochranu operačního systému před změnami.

Tabulka 3 - Přehled virtualizačních technologií, zdroj: [1]

Jak již bylo zmíněno, uvedené technologie si jsou v některých ohledech velmi podobné. Pro získání konkrétní představy o rozdílech, nebo shodnostech daných řešení, je uvedena následující tabulka porovnávající vybrané funkcionality.

Funkce	Symantec Endpoint Virtualization	Citrix XenApp	Microsoft App-V	VMware ThinApp	InstallFree Bridge
Izolace aplikací	•	✓	✓	✓	✓
Nezávislost na OS	✓	✓	✓	✓	✓
Streamování	✓	✓	✓	•	•
Integrace AD	✓	✓	✓	✓	✓
Kontrola licencí	✓	✓	✓	•	✓
Správa mezipaměti	✓	✓	✓	X	•
Agent	✓	✓	✓	X	•

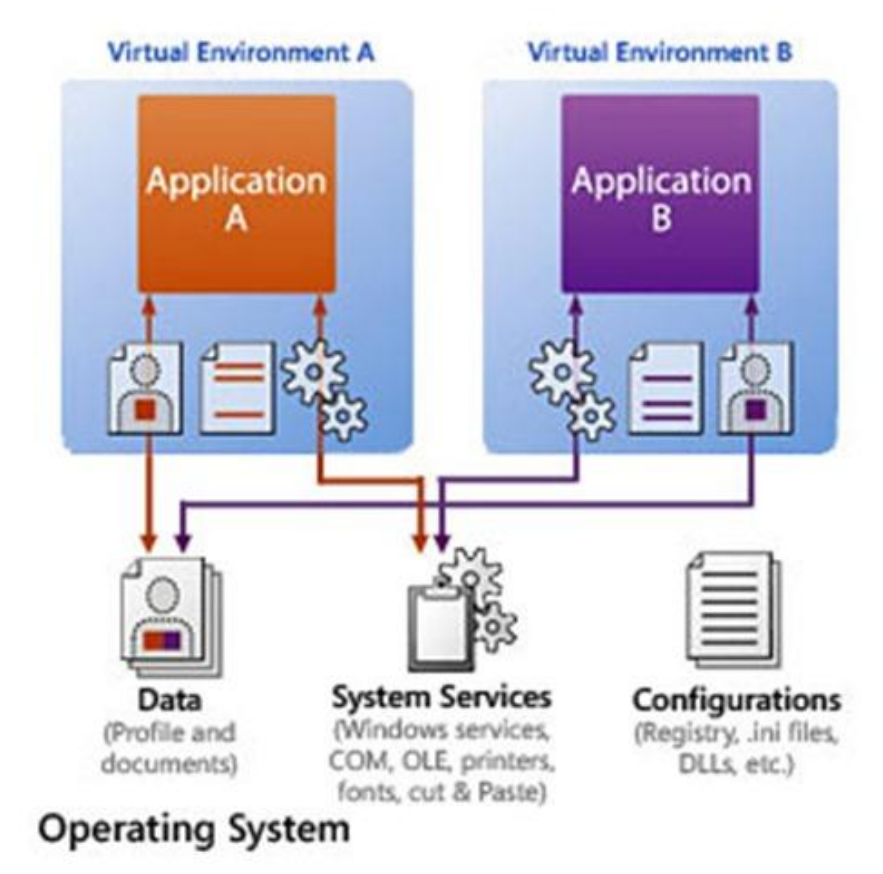
Tabulka 4 - Vlastnosti virtualizačních řešení, zdroj: [1]

- ✓ Funkce je podporována
- Funkce je podporována částečně
- X Funkce podporována není

Z tabulky lze určit skupiny podobných řešení. Jsou pouze dvě technologie (ThinApp, Bridge), které buď vůbec, nebo jen částečně, využívají koncové agenty. Stejně dvě technologie rovněž jen částečně využívají funkci streamování aplikací. Ostatní technologie mají, až na jednu výjimku, shodnou funkcionalitu. V následujících kapitolách si blíže přiblížíme technologie z obou skupin.

3.3 Microsoft Application Virtualization (App-V)

Technologie App-V je jednou z komponent Microsoft Desktop Optimization Pack (MDOP). MDOP představuje sadu technologií navržených pro podporu virtualizace uživatelského prostředí, aplikace a systému Windows. Dále umožňuje správu funkcí systému Windows. [10]



Obrázek 3 - Schéma virtuálních aplikací, zdroj: [12]

Výše uvedené schéma znázorňuje integraci souborů, služeb a nastavení do virtuálního prostředí. Tato izolace umožňuje spuštění jedné a více aplikací zároveň a dále urychluje instalaci na koncová zařízení. Využívání App-v přináší výhody oproti klasickému přístupu v podobě:

- Spouštění aplikací pomocí streamovacího serveru (není lokálně nainstalována)
- Používání navzájem nekompatibilních aplikací

- Aktualizace aplikací prostřednictvím serveru bez nutnosti konfigurace na koncové stanici
- Správa licencí aplikací (nastavení maximálního počtu současně běžících aplikací) [11, 9]

3.3.1 Komponenty APP-V

Technologie App-v je tvořena několika komponentami. V základu je lze rozdělit na část klientskou a serverovou. Klientská část je implementována na koncová zařízení a zajišťuje komunikaci se serverem. Serverová část zajišťuje správu a distribuci virtuálních aplikací. Konkrétně se App-V skládá z následujících komponent:

- Management server
- Streaming server
- Client
- Sequencer [15]

Management server

Hlavním úkolem management serveru je doručení zabalené a nakonfigurované aplikace na klientskou stanici. Server využívá Microsoft SQL Server jako datové úložiště, přičemž více serverů může využívat jeden datový zdroj. Dále ověřuje žádosti, zajišťuje bezpečnost, měření a monitoring. Také využívá AD k správě uživatelů a aplikací. Management server obsahuje dvě části. První je Managent Console. Ta slouží ke správě jednoho či více Management serveru. Umožňuje přidávání a odebírání aplikací, nastavení přístupových oprávnění pro uživatele, nebo skupiny a správu licencí. Druhou částí je Managent Web Service. Ta zprostředkovává komunikaci mezi Management Console a SQL úložištěm.[2]

Streaming server

Streaming server představuje zdroj aplikačních balíčků pro klientské stanice, které se nachází na vzdáleném pracovišti, mimo Management Server. Vzhledem k možnosti, že virtualizované aplikace mohou dosahovat velikosti až několika GB, je doporučeno umístit

datový zdroj pro koncové stanice v rámci vysokorychlostní LAN. Streamování aplikací přes WAN se vzhledem k limitující šířce pásma nedoporučuje. Server poskytuje aktualizace bez ověření požadavků skrz AD nebo SQL. Publikační služby, licencování a monitoring musí být zajištěn prostřednictvím management serveru. Streaming server představuje okleštěný nástroj k distribuci aplikací do vzdálených poboček, kde není k dispozici AD domain controller, nebo SQL sever. Management server centrálně spravuje publikování virtuálních aplikací, ale samotnou distribuci balíčků zajišťuje lokální streaming server.[2]

Client

Client je komponenta na straně pracovní stanice, která obdrží aplikace od serveru a následně zajistí jejich spuštění. Umožňuje spravovat virtuální balíčky. Lze nastavit, zda se aplikace celá stáhne a bude přístupná v režimu offline. Obsahuje nastavení velikosti lokální paměti, kterou si aplikace může alokovat. Dále lze nastavit frekvenci dotazů na případné aktualizace na straně serveru.[2]

Sequencer

Tento program je využíván k vytváření aplikačních balíčků, které jsou posléze distribuovány na koncová zařízení. Aplikace se instaluje na desktop, na kterém je posléze spuštěna instalace virtualizované aplikace. Sequencer zaznamenává proces instalace a vytváří podklady pro virtualizované aplikace. Výstupem aplikace je aplikační balíček, který je tvořen následujícími soubory:

- Sft- soubor obsahující veškerá přenesená data během instalace
- Osd- jedná se o odkaz na sft soubor, který obsahuje veškeré informace potřebné k načtení a spuštění aplikace
- Ico- obsahuje ikony spustitelných souboru
- Manifest.xml- využívá se k distribuci aplikací pomocí systému na doručení elektronického softwaru (ESD)
- Sprj- jedná se o projektový soubor zaznamenávající data pro budoucí změny v aplikaci

Aplikace může vytvářet také soubory typu msi. Tyto soubory jsou využívány k nasazení aplikací na koncové stanice v módu stand-alone (bude popsán v následující kapitole).[5]

3.3.2 App-V módy

App-V lze provozovat ve dvou režimech. První režim je stand-alone. Tento režim je využíván zejména u zařízení, která nejsou aktuálně připojena k doméně, nebo na vzdálených pracovištích, kde není provozován streamovací server. Tento režim využívá pouze Sequencer k vytváření instalací virtualizovaných aplikací, nasazovaných na koncová zařízení. Samotné nasazení lze provést prostřednictvím centrální správy aplikací, nebo lze aplikaci nasadit z CD, nebo USB disku.

Streamovací mód představuje efektivnější způsob správy aplikací. Zajišťuje rychle nasazení a aktualizace spravovaných aplikací. Tento mód využívá jak Sequencer, tak i streaming (management) server. V případě, že není využit management server, oprávnění je řízeno prostřednictvím Acces Control Listu umístěném v SQL databázi. [2]

3.4 ThinApp

ThinApp je virtualizační řešení od společnosti VMware. Hlavní předností tohoto řešení je, že aplikace lze provozovat bez samotného klienta. Toho je docíleno integrací klienta do balíčku aplikace. Tím je zajištěna přenositelnost aplikací. Aplikace jsou kompatibilní jak s 32bitovou, tak i 64bitovou architekturou Windows. Systém běží na systému Windows NT a novějším. Systém neobsahuje distribuční mechanismus, ale aplikace jsou připraveny na streamování hned po vytvoření. Nasazení aplikací může probíhat pomocí klasických distribučních systémů. Také lze využít funkcí AD. Dále lze distribuci zajistit pomocí sdílených položek, nebo instalační webové stránky.

Aktualizace aplikací je realizována nahrazením stávající aplikace. K aktualizacím účelům slouží funkce AppSync. Ta má za úkol automaticky aktualizovat aplikace nacházející se na koncových stanicích. Aktualizační funkce je aktivována pomocí odkazu, který je nutné integrovat do samotného balíčku aplikace.

System existuje ve dvou edicích:

- Starter
- Enterprise

První edice představuje základní nástroj pro vytváření virtuálních aplikací. Je vhodná spíše pro malé firmy, jelikož neobsahuje mnoho funkcí. Edice Enterprise oproti tomu nabízí řešení pro větší firmy. Obsahuje možnost zabezpečení virtuálních aplikací. Integruje aktualizací funkci. [1]

3.4.1 Vytváření balíčku

Proces virtualizace aplikací začíná pomocí programu Setup Capture. Ten vytváří distribuční balíček, který obsahuje všechny potřebné komponenty aplikace a administrátorská nastavení. Během procesu virtualizace je vytvořena projektová složka, kde je umístěna veškerá konfigurace. Odtud je možné vytvářet aktualizace a případné změny konfigurace, které mají za následek vytvoření dalšího samostatného balíčku. Konfigurační nastavení popisující způsob aktualizace, aplikační nastavení a integrace s lokálním operačním systémem jsou integrovány do aplikačního balíčku během jeho vytváření. Výstupem virtualizace, kromě klasických „exe“ založených balíčků, může být i MSI balíček.

Vytváření virtuálních aplikací je doporučováno provádět na počítači s čistou instalací. Tím je zajištěno, že všechny soubory a komponenty nezbytné pro vizualizovanou aplikaci jsou detekovány. Dále je vhodné použít nejstarší verzi operačního systému, na kterém je plánováno virtualizované aplikace provozovat.

Během procesu vytváření je možné nastavit zabezpečení balíčku. Lze specifikovat, kdo bude mít k vytvořené aplikaci přístup. Oprávnění lze přidat prostřednictvím AD. Mechanismus přístupu je pro koncového uživatele skrytý a nelze ho před spuštěním aplikace nijak modifikovat. Zabezpečení je aktivní i po přesunutí na jiná koncová zařízení.[7]

3.4.2 Nasazení balíčku

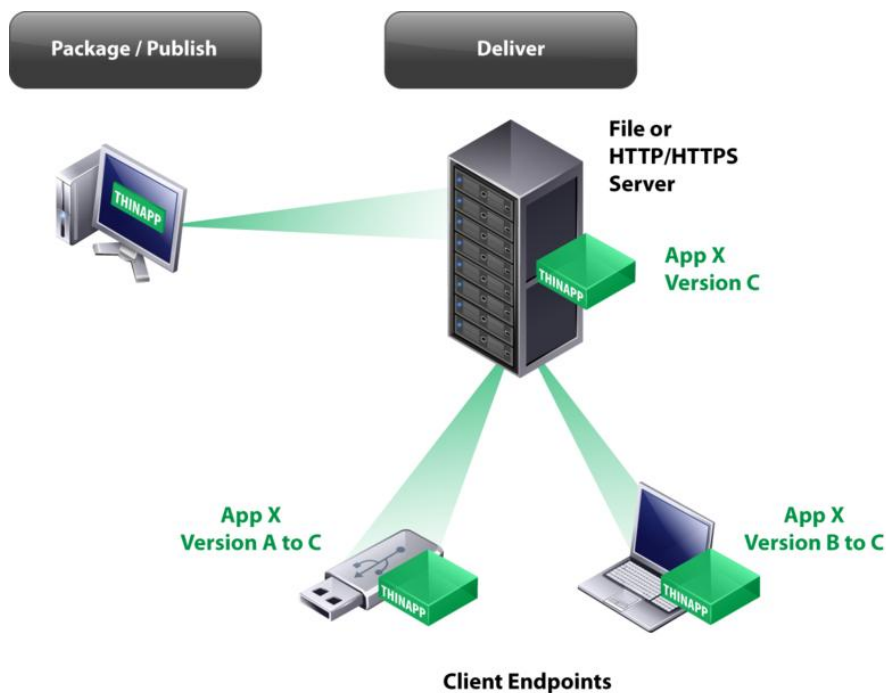
Proces nasazení je jednodušší oproti klasické instalaci, jelikož interakce aplikace a operačním systémem je minimální. Existují dva způsoby nasazení aplikace. První je

centralizovaný a druhý je decentralizovaný. Tyto dva přístupy jsou závislé na způsobu spuštění balíčků. Stejný balíček může být nasazen v dvou různých spouštěcích módech:

- Streaming mode
- Deployed mode

Streaming mód má centrálně uloženou aplikaci, ke které přistupuje více uživatelů. Tento mód je označován one-to-many a zastupuje centralizovaný způsob nasazování a aktualizování aplikací. Tento způsob je vhodný pro firemní řešení, kde jsou počítače stále online. Aplikace je inicializována pomocí zástupce na ploše, který ji zpustí na vzdáleném úložišti. Virtualizovaná aplikace je streamována do lokální paměti, podle toho, co je zrovna vyžadováno. Právě nutnost připojení koncových stanic do sítě je hlavní nevýhoda. Dále je nutné zajistit robustní síťové připojení v případě většího počtu koncových zařízení.

V druhém režimu je aplikační balíček nejprve nasazen na koncový systém, až poté je aplikace přístupná z místního zařízení. Koncový uživatel spouští aplikaci z lokální stanice, což umožňuje práci offline. Toto řešení podporuje distribuci aplikací jak na fyzické, jak na virtuální koncové stanice. Aplikace může být uložena na místním disku, nebo na externím USB zařízení. Nasazení aplikačního balíčku může probíhat pomocí AD, softwaru třetích stran, nebo pomocí CD či USB zařízení. [7]



Obrázek 4 - Schéma distribuce ThinApp, zdroj: [13]

3.5 Výběr vhodné technologie

Pro výběr vhodné technologie byla zvolena Saatyho metoda, která slouží k stanovení vah kritérií. Princip spočívá v preferenčním vztahu dvojic kritérií, kdy se kromě směru preference určuje také velikost. Velikost preferencí je dána bodovou stupnicí. Do této metody byly zahrnuty celkem tři relevantní technologie. Kritéria vycházejí z přehledu identifikovaných technologií. Výsledky jsou zobrazeny v následující tabulce. [13]

Kritérium		Preference		
Název	Váha	XenApp	App-V	ThinApp
Streamování	0,3	0,45	0,45	0,1
Správa licencí	0,1	0,54	0,32	0,14
Správa aplikací	0,3	0,31	0,58	0,11
Správa mezipaměti	0,05	0,47	0,43	0,1
Použití agenta	0,15	0,13	0,26	0,61
Vytváření balíčků	0,1	0,27	0,46	0,27
Výsledek		0,35	0,45	0,20

Tabulka 5 - Porovnání technologií Saatyho metodou, zdroj: vlastní pořízení

Z výsledků je patrné že nejlepšími hodnotami dosáhla technologie App-V. Následuje XenApp a poslední je ThinApp. Na základě těchto výsledků bude v praktické části práce implementována technologie App-V.

4 Vlastní řešení

V praktické části práce je popsán postup implementace technologie App-V, která byla stanovena v teoretických východiscích jako nejvhodnější. Výhody zvoleného řešení, stejně jako princip fungování jsou popsány v teoretické části, a proto v případě potřeby bude odkazováno na příslušné kapitoly. Postup implementace je rozdělen do třech logických částí. Jedná se o následující prvky:

- Sequencer
- Server
- Klient

Tyto logické části jsou členěny do shodné struktury postupu, která by měla přispět k zpřehlednění obsahu. V každé kapitole jsou definovány předpoklady, které jsou nezbytné pro správnou funkčnost. Dané předpoklady svým rozsahem přesahují rámec této diplomové práce, a proto jim nebude věnována větší pozornost. Další částí je postup instalace jednotlivých komponent a případná konfigurace. Zde je kladen důraz na názornost prostřednictvím pořízených obrazových materiálů (screenshotů). Ty znázorňují nejdůležitější kroky instalace a dokreslují její průběh. Poslední část je věnována samotnému používání jednotlivých komponent. To je znázorněno na konkrétním případu užití. Takto popsaná struktura přináší jako celek přehledný obraz o implementaci a následném použití technologie APP-V.

Před samotným popisem postupu implementace je nutné definovat prostředí, v kterém bude tato technologie provozována. Jak již bylo popsáno v teoretické části této práce, jedná se o technologii klient-server. Ta jako taková předpokládá kvalitní síťovou infrastrukturu. Pro efektivní využití této technologie je dále předpokládáno větší množství koncových stanic (v našem případě 150 stanic), které jsou propojeny v rámci jedné domény. Z výše uvedeného je patrné, že toto řešení je vhodné pro střední, až spíše velké firmy. Dále je možné tuto technologii využít na středních, případně vysokých školách. V našem konkrétním případě se jedná o firmu, která se zabývá finančnictvím. Vzhledem k charakteru zaměření firmy je jedinou možnou nevýhodou zvýšená úroveň zabezpečení oproti běžnému provozu. Toto omezení by však nemělo znamenat významnější překážku při implementaci zvolené technologie.

4.1 Sequencer

První komponenta, která je v rámci praktické části popsána se nazývá Sequencer. Jak bylo popsáno v teoretické části, jedná se o aplikaci, která vytváří softwarové balíčky, které jsou pomocí serveru distribuovány ke koncovým stanicím. Díky těmto balíčkům jsou koncové stanice schopny spouštět příslušné aplikace, aniž by je museli nejprve instalovat. To je základní princip fungování. Aplikace Sequencer je blíže popsána v kapitole 2.2.1.

4.1.1 Předpoklady a systémové požadavky

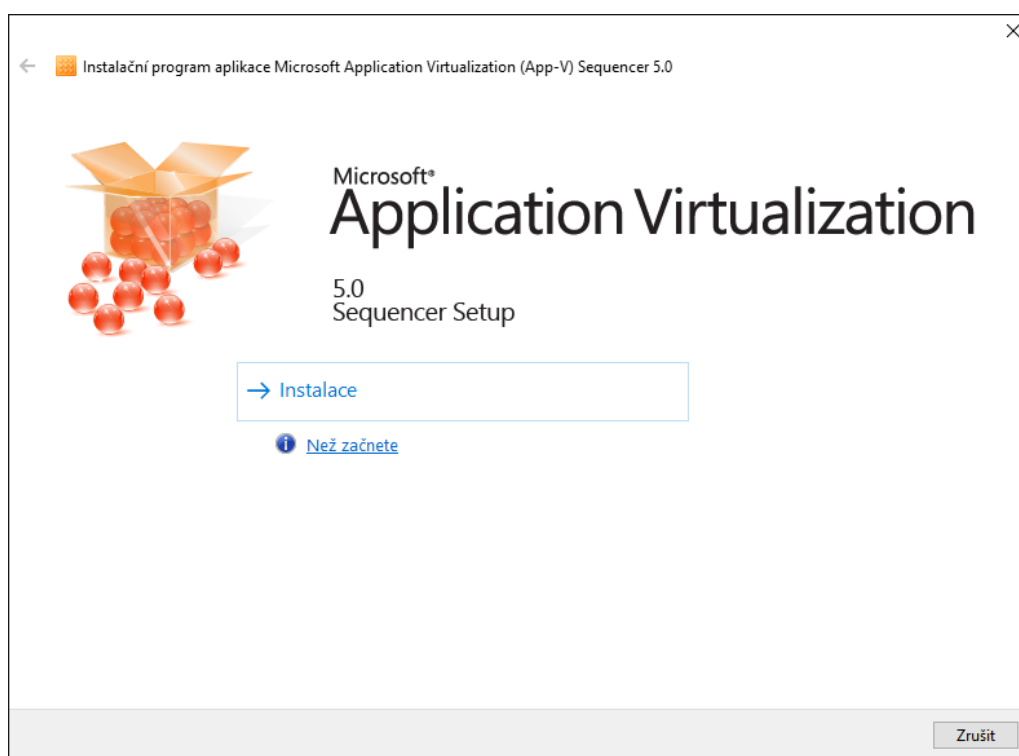
Nejprve si stanovíme předpoklady pro úspěšnou implementaci této aplikace. Aplikace sequencer bude instalována na virtuální PC. Pro vytvoření virtuálního PC bude využita aplikace Virtual Box. Aplikace je instalována na virtuální PC z důvodu možnosti vytvářet snapshoty. Ty zaznamenávají stav počítače před určitou změnou. Tato vlastnost bude využívána při vytváření virtualizačních balíčků. Nejprve bude zaznamenán stav virtuálního stroje. Poté bude provedena instalace zvolené aplikace, během které se vytvoří virtuální aplikace. Na konec bude vrácen stav stroje do stavu před instalací. Tento proces se opakuje při vytváření každé virtuální aplikace, aby bylo dosaženo co nejlepších výsledků. Na virtuálním PC bude provozován operační systém Windows 7 Profesionál (32bit), jelikož je tento systém provozován i na koncových stanicích. Je doporučována nová instalace bez dalších komponent a aktualizací. Dále je doporučováno vypnutí služeb Defender, Windows search a Windows update, které by mohli ovlivnit stav počítače a následný proces vytváření virtuální aplikace.

Minimální systémové požadavky jsou dnešními standardně vybavenými stanicemi značně překonávány. Aplikaci lze provozovat na běžném kancelářském počítači. Minimální požadavky aplikace jsou následující:

- Procesor Intel Pentium III, 850 MHz
- RAM – 256MB, 500 MB stránkový soubor
- Dva fyzické disky každý s kapacitou 20 GB (případně jeden fyzický disk rozdělen na dvě partišny)
- Operační systém - Windows 7 (SP1), Windows 8

4.1.2 Instalace

Při prvním spuštění instalačního souboru aplikace Sequencer jsme informováni o chybějícím .NET Frameworku. Proto je před samotnou instalací aplikace nutné nainstalovat rozhraní .NET Framework 4.0. Jedná se o standardní instalaci bez nutnosti konfigurace. Další komponenta, která je vyžadována před instalací aplikace Sequencer je PowerShell. V našem případě je instalován ve verzi 3.0. Instalace stahuje dodatečné soubory a z toho důvodu je nutné mít virtuální PC připojeno k internetu (po instalaci je již PC odpojeno od internetu). Po instalaci obou komponent se již spustí samotná instalace aplikace Sequencer.



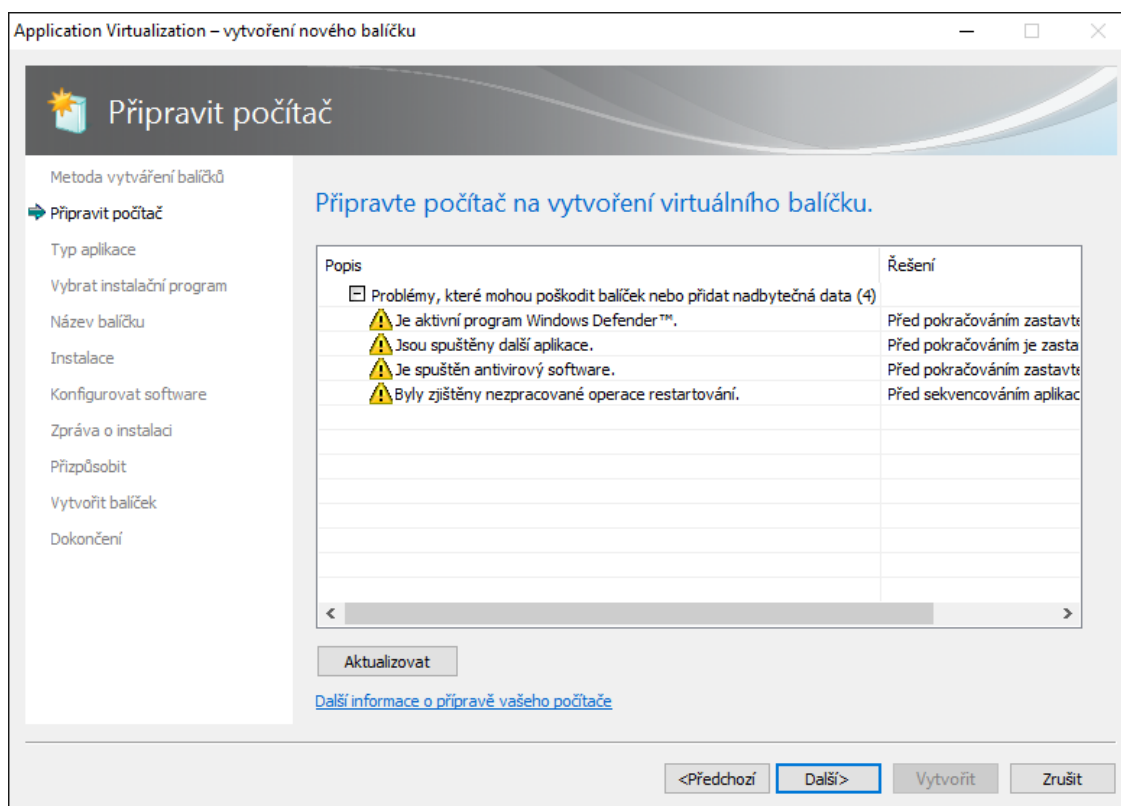
Obrázek 5 - Úvodní instalační okno, zdroj: vlastní pořizení

Následují standardní kroky v podobě licenčních ujednání a programu pro zlepšování softwaru. Poté již stačí potvrdit instalaci a aplikace se bez dalších konfigurací doinstaluje. Instalační proces je velmi snadný a zabere několik minut.

4.1.3 Vytváření aplikačních balíčků

Po instalaci je možné ihned s aplikací začít pracovat. Po spuštění se zobrazí úvodní okno se základní nabídkou. Je zde možnost vytvoření nového balíčku virtuální aplikace, případně upravit již existující. Úprava stávajícího balíčku je rozdělena na tři části. Lze aktualizovat aplikaci v existujícím balíčku. Dále upravit vlastnosti existujícího balíčku (nastavení nasazení). Nebo lze přidat novou aplikaci do stávajícího balíčku (aplikace typu plug-in a podobně).

Následující postup je věnován způsobu vytvoření nové virtuální aplikace. Při vytváření aplikace lze vybrat ze dvou metod. První metoda vytváří balíček aplikace při standardní instalaci, kdy App-V monitoruje všechny změny během instalace. Druhá volba vytváří balíček pomocí akcelerátoru balíčků, který používá instalační soubory aplikace k automatickému vytvoření balíčku. V tomto kroku je zvolena první metoda klasické instalace, která je také výchozí volbou. V dalším kroku je připravován počítač na samotnou virtualizaci. Zde jsme informováni o nedostacích na pracovní stanici.



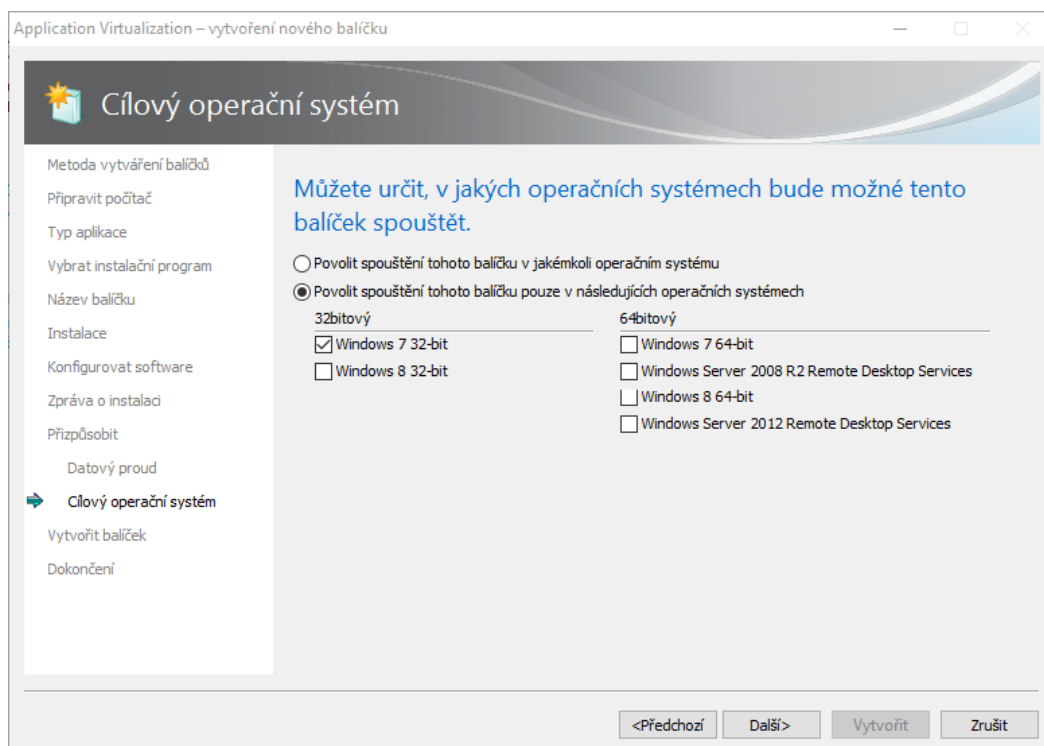
Obrázek 6 - Příprava PC, zdroj: vlastní pořízení

Z obrázku výše lze identifikovat čtyři problémy na PC. Nebyla ukončena služba Windows Defender, je spuštěna další aplikace, běží antivirový program a je vyžadován restart PC. Po restartování stanice a ukončení zmíněných služeb a aplikací lze bez problému pokračovat v procesu.

V dalším kroku se specifikuje typ vytvářeného balíčku aplikace. Lze vybrat ze tří možností:

- Standardní aplikace
- Doplněk nebo modul plug-in
- Middleware

Middleware vytváří balíčky pro software rozhraní, které je využíváno standardními balíčky. Druhá možnost umožňuje vytvořit balíček, který rozšiřuje funkce standardních aplikací. Poslední možností, která bude také zvolena, je vytvoření balíčku se standardní aplikací. Následuje zvolení cesty k instalačnímu souboru požadované aplikace (EnterpriseArchitect) a pojmenování balíčku virtuální aplikace. Poté je již spuštěna instalace samotné aplikace. Instalace probíhá stejným způsobem jako při klasické instalaci aplikace. Po úspěšné instalaci je nutné informovat monitorovací systém o dokončení. Následující konfigurační krok umožňuje otestovat a nakonfigurovat nainstalovanou aplikaci. Po úspěšném spuštění a konfiguraci aplikace lze optimalizovat možnosti samotného balíčku. Jedná se o přizpůsobení se v případě pomalých sítí. Dále lze specifikovat operační systém, na kterém aplikace bude provozována. V tomto případě byl balíček nakonfigurován tak, aby ho bylo možné spustit pouze na Windows 7, 32bit (to je standardní konfigurace použitých koncových stanic).



Obrázek 7 - Specifikace cílového OS, zdroj: vlastní pořízení

V poslední části lze přidat popis k vytvářenému balíčku a potvrdit možnost vytvoření. Sequencer vytvoří složku, do které umístí zdrojové soubory virtualizované aplikace. Zejména se jedná o soubory „EA.appv“ a „EA.msi“, které jsou využívány při nasazení na koncové stanice. V tomto případě bude využit soubor „EA.appv“, který bude distribuován pomocí streamovacího serveru na koncová zařízení. Instalace serverové části a následná distribuce virtualizované aplikace je popsána v následující kapitole.

4.2 Serverová část

Serverová část má na starosti správu a distribuci virtuálních aplikací. Nejprve importuje virtuální aplikaci vytvořenou v aplikaci Sequencer. Posléze lze aplikační balíčky modifikovat. Lze specifikovat přístupová oprávnění, vytvářet skupiny aplikací a mnohé další. Když je aplikační balíček nakonfigurován, je ho možné publikovat, což umožní uživatelům začít s aplikací pracovat. Správu balíčků a jejich distribuci zajišťují odlišné komponenty. Management server spravuje nastavení balíčků a využívá funkcí MS SQL serveru, který je nezbytnou součástí. Oproti tomu publikační server provádí streamování aplikací na koncová zařízení. Blíže jsou tyto komponenty popsány v kapitole 2.2.1.

4.2.1 Předpoklady a systémové požadavky

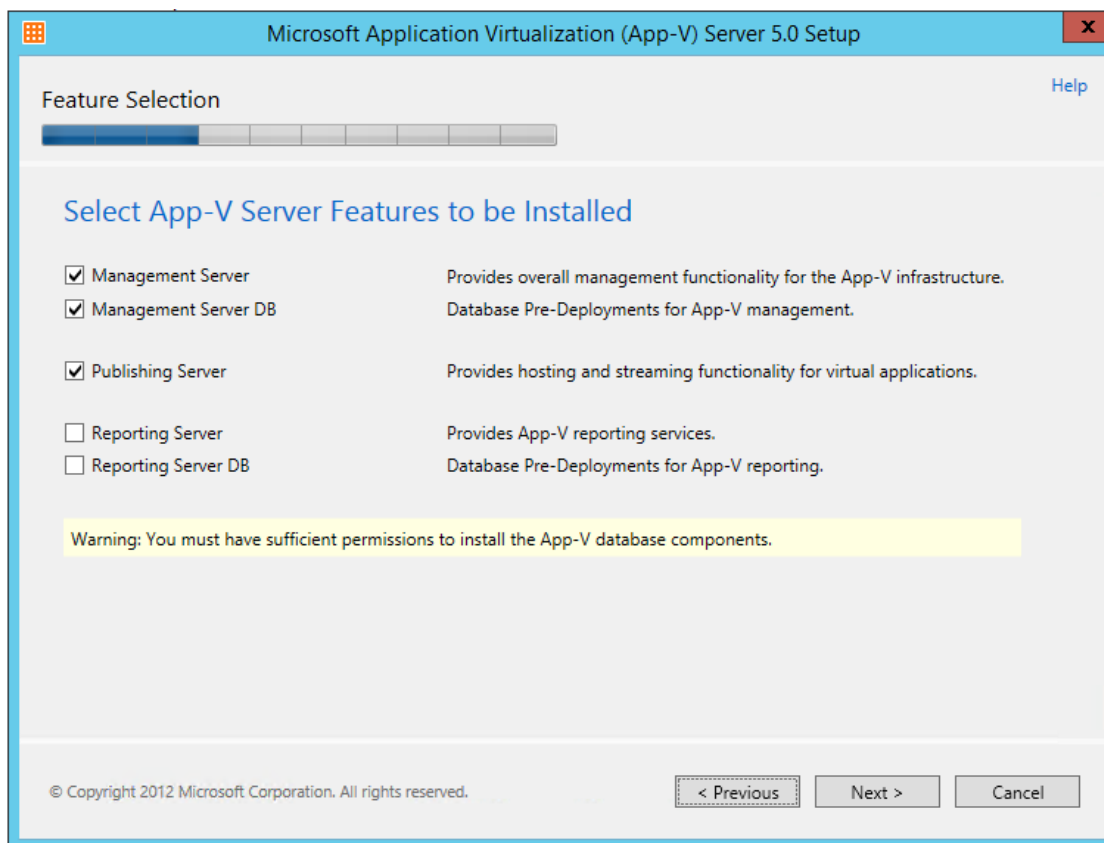
Pro nasazení serverové části bude využit Windows server 2012 R2. Tento server a všechny koncové stanice jsou zařazeny do podnikové domény. V rámci podnikové domény jsou vytvořeny skupiny administrátorů a běžných uživatelů. Jak již bylo zmíněno, nedílnou součástí zvoleného řešení je MS SQL server. Jedná se plnohodnotný SQL server, který je v tomto případě nainstalován ve verzi 2008+. Instalace a konfigurace výše zmíněných součástí překračuje rámec této diplomové práce a proto zde nebude blíže popsána.

Dále jsou specifikovány minimální požadavky na systém:

- Procesor – 1,4 GHz
- RAM - 2 GB
- Místo na disku - 200 MB
- Operační systém - Windows Server 2008 R2 (SP1 nebo novější), Windows Server 2012 (R2)

4.2.2 Instalace a konfigurace

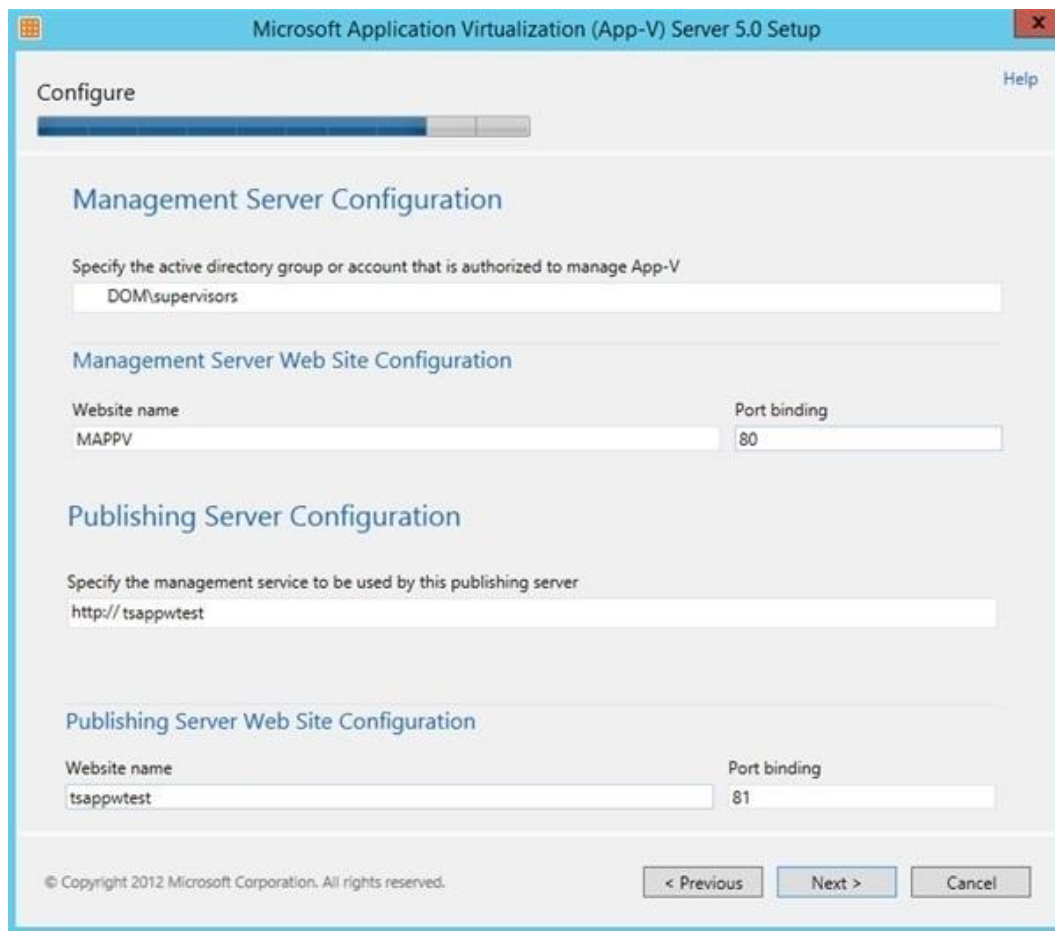
Stejně jako v případě instalace aplikace Sequencer, tak i v tomto případě je při spuštění instalace nejprve vyžadováno rozhraní .NET Framework 4.0. Po instalaci všech zmíněných komponent je možné spustit instalaci App-V Serveru. Úvodní obrazovka obsahuje základní informace o verzi, dokumentaci a licenci. V další části se vybírají jednotlivé komponenty, které budou zahrnuty do instalace. Výsledné nastavení je znázorněno na následujícím obrázku.



Obrázek 8 - Výběr komponent App-V Serveru, zdroj: vlastní pořízení

Z obrázku je patné že do instalace jsou zahrnuty komponenty Management Server (DB) a Publishing Server. První zmiňovaná obsahuje funkce pro správu App-V infrastruktury. Druhá obstarává funkce pro streamování virtuálních aplikací. Naopak Reporting Server není pro účely této práce nezbytnou komponentou.

Následuje vytvoření a konfigurace Management Server Database. Zde jsou konfigurační nastavení ponechána ve výchozím nastavení. Instalace App-V Serveru začíná komunikovat s SQL serverem. Je potřeba dát na zřetel, že je nutné mít právo zápisu do SQL databáze, jinak nebude možné v instalaci pokračovat. Nastavení Management a Publishing Serveru je znázorněno na následujícím obrázku.



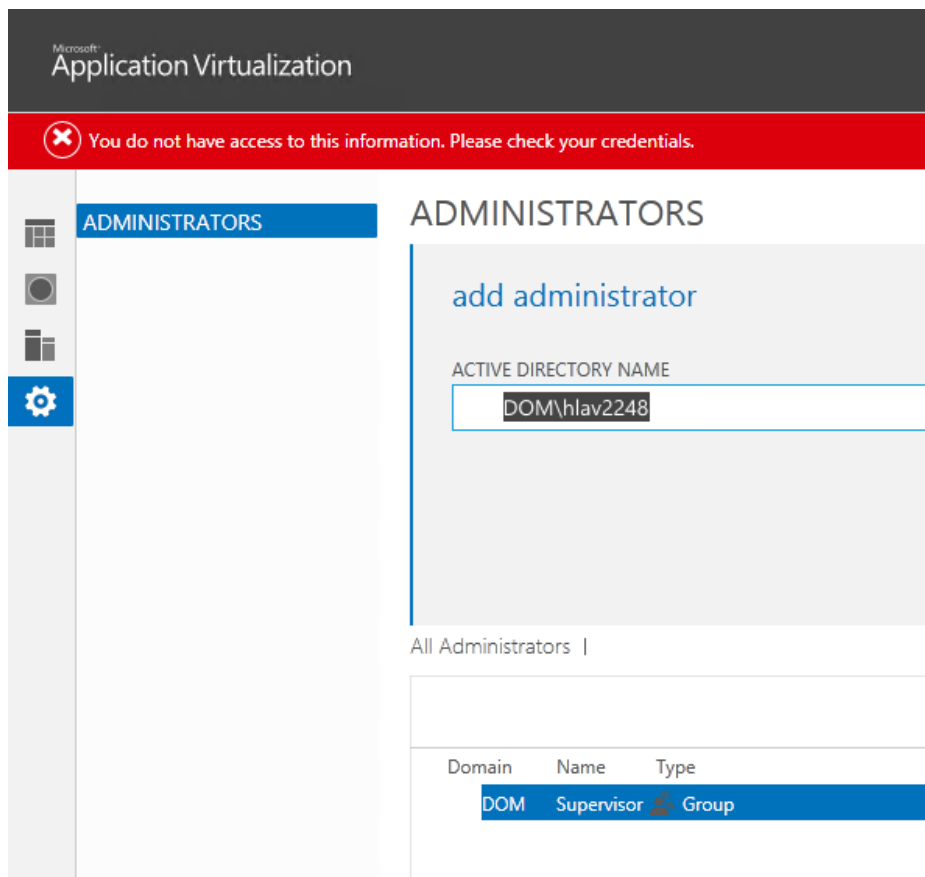
Obrázek 9 - Nastavení Serveru App-V, zdroj: vlastní pořízení

Nejprve je nutné specifikovat skupinu uživatelů, kteří budou mít přístup do správy App-V. V tomto případě to budou členové skupiny „supervisors“, v rámci domény „DOM“. Dále byl nastaven název webové stránky Management serveru a příslušný port. V rámci Publishing serveru byla adresa služby pro správu nastavena na „tsappwtest“. Publishing Server byl pojmenován stejným názvem a byl mu přiřazen port 81.

V posledním kroku je provedena rekapitulace instalovaných komponent. Instalujeme tedy Management Server a k němu přidruženou databázi. Další instalovanou komponentou je Publishing Server. Poté stačí potvrdit instalaci a instalační proces se již sám dokončí.

Serverová administrace je dostupná skrz webové rozhraní. Adresa vychází z nastavení, které bylo provedeno v průběhu instalace. Celá adresa je ve formátu „*http://tsappwtest/console.html*“. Při prvním spuštění se vyskytli problémy s přístupovými právy. Během instalace byl špatně zadán název skupiny uživatelů mající přístup do

konzole. To pravděpodobně způsobilo níže uvedenou chybu. I přes přidání správné skupiny nebyl problém odstraněn. Problém byl vyřešen až přidáním konkrétního uživatele do seznamu administrátorů.



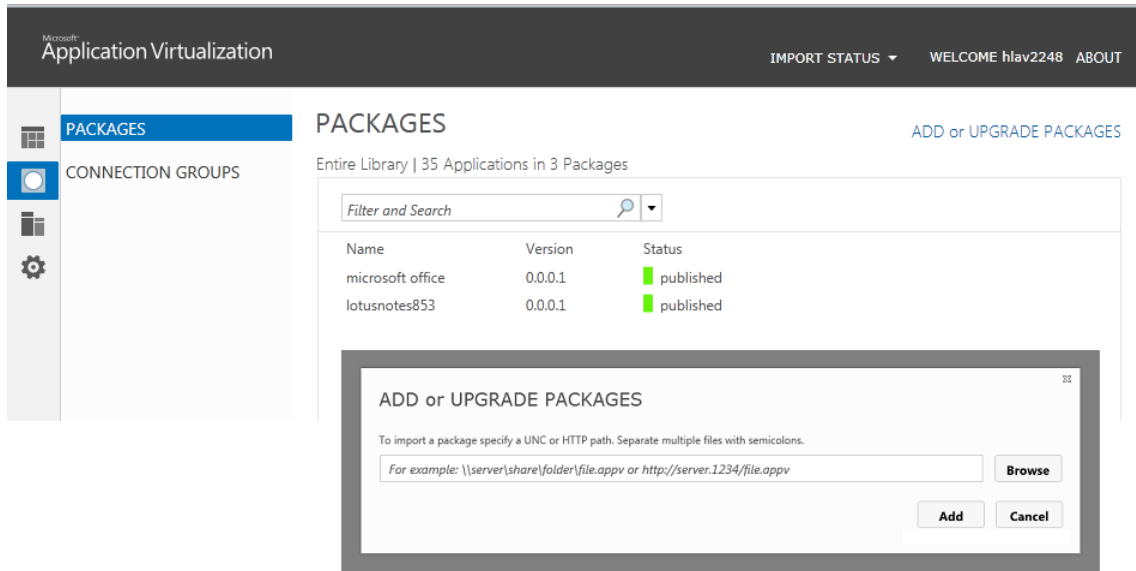
Obrázek 10 - První spuštění administrátorské konzole, zdroj: vlastní pořízení

4.2.3 Správa a distribuce virtuálních aplikací

Menu konzole je rozděleno do čtyř základních částí. Úvodní stránka má pouze informační charakter a neobsahuje žádné volby. Následuje správa balíčků, která představuje nejdůležitější část. Zde je možné spravovat samotné aplikační balíčky, ale i vytvářet skupiny. Další záložka obsahuje přehled nainstalovaných serverů. Poslední záložka je správa administrátorů. Zde lze přidávat jednotlivé správce, popřípadě celé skupiny v rámci AD.

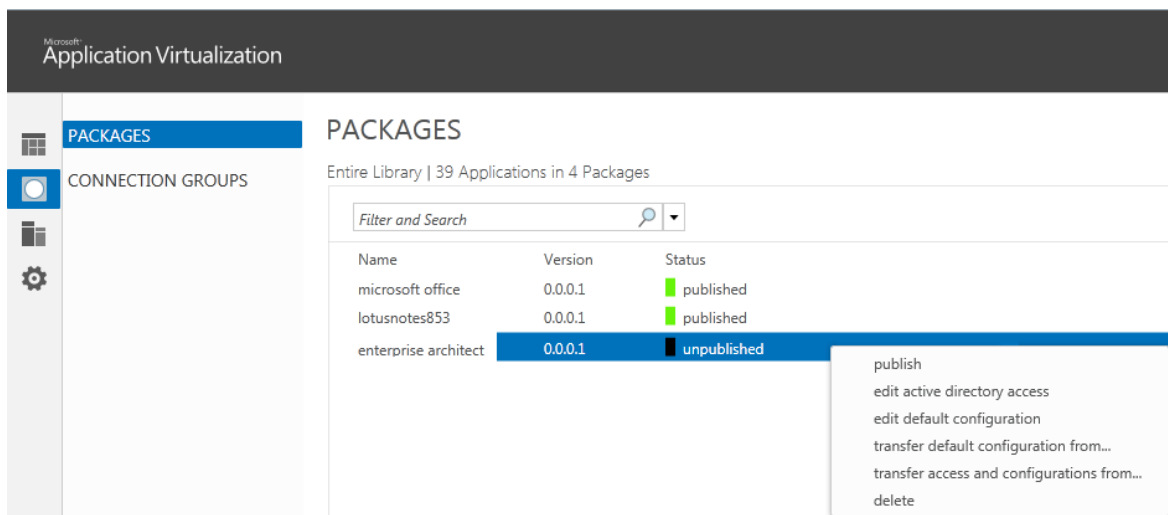
Přidání nově vytvořeného balíčku se provádí v sekci „packages“. Jsou zde všechny již importované balíčky. Zvolíme možnost „ADD or UPGRADE PACKAGES“. Systém si vyžádá cestu k nově vytvořenému balíčku. Cestu lze zadat ve formátu UNC a HTTP.

Zadáme cestu k souboru, který jsme vytvořili v aplikaci Sequencer a posléze přenesli na server. Poté již stačí potvrdit tlačítkem přidat. Průběh přidání balíčku je znázorněn na následujícím obrázku.



Obrázek 11 - Správa aplikačních balíčků, zdroj: vlastní pořízení

Aplikační balíček je posléze přidán do seznamu aplikací. V tuto chvíli ještě není aplikace dostupná pro koncová zařízení. Je potřeba ještě provést konfiguraci. V konfiguraci lze nastavit přístupová oprávnění pro danou aplikaci. Lze nastavit výchozí nastavení, které obsahuje například přidání či odebrání jednotlivých částí programu a možnosti vytvoření zástupce na ploše. Toto nastavení lze převzít z již nastavených balíčků. Na následujícím obrázku je znázorněn přidávaný balíček, který ještě nebyl publikován. Před samotným publikováním nastavíme přístup pomocí volby „edit active directory access“. Přidání přístupu pro uživatele, kteří jsou členové skupiny „Domain Users“, bude vypadat následovně: „*DOM\ Domain Users*“. Na začátku je definována doména, oddělená zpětným lomítkem, za kterým následuje AD skupina. Nastavení je potřeba ještě potvrdit.



Obrázek 12 - Konfigurace aplikačního balíčku, zdroj: vlastní pořízení

Konzole dále umožňuje specifikovat vlastní skupiny aplikací. Nově vytvořené skupině se nadefinují přístupová práva stejným způsobem, jako tomu bylo u samotných balíčků. Do takto vytvořené skupiny stačí už jen přidat požadované aplikační balíčky. Posledním krokem je publikování aplikačního balíčku, případně celé skupiny balíčků. V možnostech požadovaného balíčku či skupiny stačí zvolit možnost „publish“ a aplikace je během chvíle dostupná pro koncové uživatele.

4.3 Klientská část

Poslední součástí App-V, kterou je potřeba implementovat, je klientská část. V předchozích kapitolách byl popsán proces virtualizace pomocí aplikace Sequencer. Virtualizovaná aplikace byla posléze importována na server, kde byla nakonfigurována a publikována. Klient zprostředkovává komunikaci se serverem a zajišťuje stahování a aktualizaci publikovaných aplikací. Dále klient zajišťuje samotné spuštění virtuálních aplikací na koncové stanici.

4.3.1 Předpoklady a systémové požadavky

Instalace klientské části je realizována na operačním systému Windows 7 (32 bit). Tento systém je využíván na většině koncových zařízení v daném prostředí. Koncové stanice jsou součástí domény, ve které je provozována i serverová část. Před samotnou instalací klientské aplikace jsou vyžadovány podpůrné komponenty. Nezbytnou komponentou je rozhraní .NET Framework 4.0. Další komponenta, která musí být na

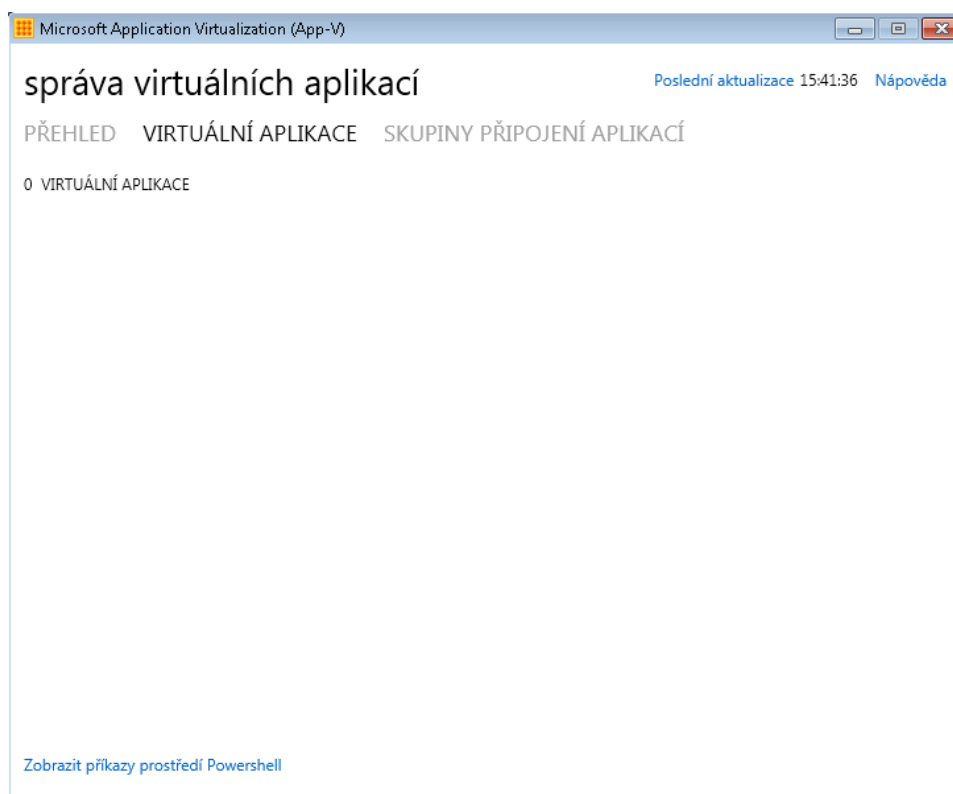
koncové stanici nainstalována je Powershell 3.0. Instalace je prováděna formou aktualizací. Z tohoto důvodu je doporučeno mít koncovou stanici připojenou k internetu, kdyby byla vyžadována některá chybějící aktualizace.

Klientská aplikace má následující systémové požadavky:

- Procesor – 1,4 GHz
- RAM – 1 GB
- Volná paměť na disku – 100 MB (nezahrnuje prostor pro virtuální aplikace)
- Operační systém – Windows 7 SP1, Windows 8

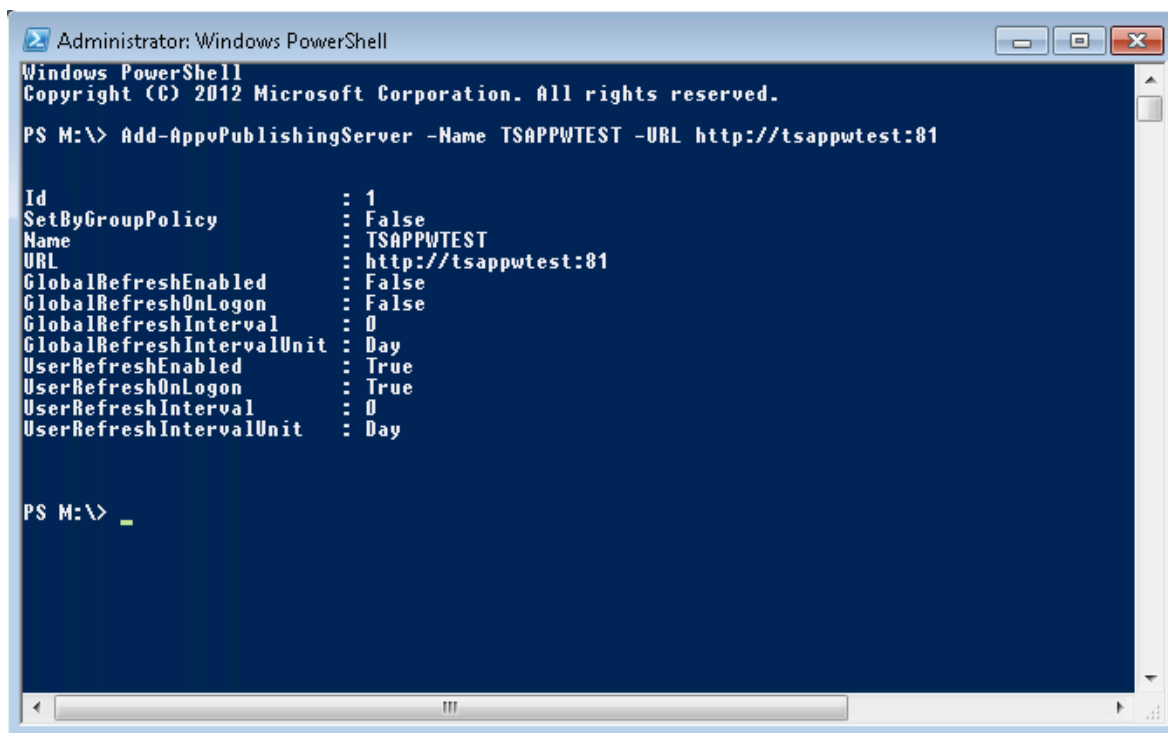
4.3.2 Instalace a konfigurace

Po nainstalování zmíněných komponent a požadovaných aktualizací je možné spustit samotnou instalaci klienta. Během instalace není vyžadováno žádné nastavení. Stačí pouze potvrdit a aplikace se sama nainstaluje. Aplikaci lze spustit hned po dokončení instalace. Při prvním spuštění klient neobsahuje v seznamu aplikací žádné virtuální aplikace, ani nekomunikuje se serverem.



Obrázek 13 - První spuštění klienta, zdroj: vlastní pořízení

Klienta je třeba nakonfigurovat prostřednictvím konzole PowerShell. Nejprve je nutné přidat publikační server. K tomu slouží příkaz „Add“. Jako parametry předáme název, adresu a port, na kterém je publikační server spuštěn. Kompletní příkaz je uveden na následujícím obrázku.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

PS M:\> Add-AppvPublishingServer -Name TSAPPWTEST -URL http://tsappwtest:81

Id                : 1
SetByGroupPolicy  : False
Name              : TSAPPWTEST
URL               : http://tsappwtest:81
GlobalRefreshEnabled : False
GlobalRefreshOnLogon : False
GlobalRefreshInterval : 0
GlobalRefreshIntervalUnit : Day
UserRefreshEnabled : True
UserRefreshOnLogon : True
UserRefreshInterval : 0
UserRefreshIntervalUnit : Day

PS M:\> _
```

Obrázek 14 - Přidání publikačního serveru, zdroj: vlastní pořízení

Po přidání publikačního serveru se nám zobrazí výpis nastavení. V něm si je možné všimnout názvu a adresy přidávaného publikačního serveru. Kromě těchto informací nás dále zajímá nastavení proměnných „GlobalRefreshEnabled“ a „GlobalRefreshOnLogon“. První proměnná udává možnost aktualizovat klienta prostřednictvím serveru. Druhá proměnná specifikuje aktualizování klienta po spuštění klienta. Obě tyto proměnné jsou v základním nastavení nastaveny na hodnotu „False“ a proto nemohou být stahovány virtuální aplikace. Pro správné fungování klienta je nutné obě tyto proměnné nastavit na hodnotu „True“. K tomu slouží následující dva příkazy:

- *Get-AppvPublishingServer | Set-AppvPublishingServer - GlobalRefreshEnabled \$true*
- *Get-AppvPublishingServer | Set-AppvPublishingServer - GlobalRefreshOnLogon \$true*

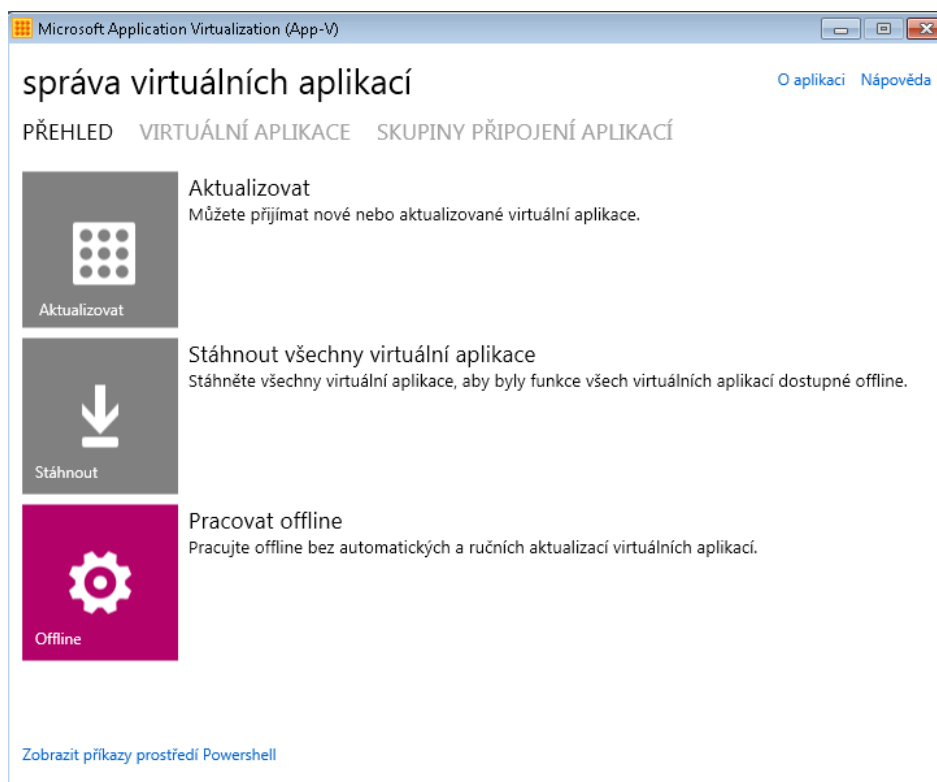
Prvním příkazem povolíme aktualizování ze serveru. Druhý příkaz zajistí aktualizaci virtuálních aplikací při spuštění. Konfigurace klienta je těmito kroky již hotová a klienta je možné začít používat. Pro urychlení procesu synchronizace klienta se serverem, lze spustit synchronizační procesor ručně zadáním následujícího příkazu.

- *Sync-AppvPublishingServer -Name TSAPPWTEST*

Příkaz pouze zahájí synchronizaci s námi definovaným publikačním serverem. Synchronizace zabere v závislosti na dané síťové infrastruktuře, množství a velikosti virtuálních aplikací řádově několik minut. Poté je již možné začít pracovat s virtuálními aplikacemi.

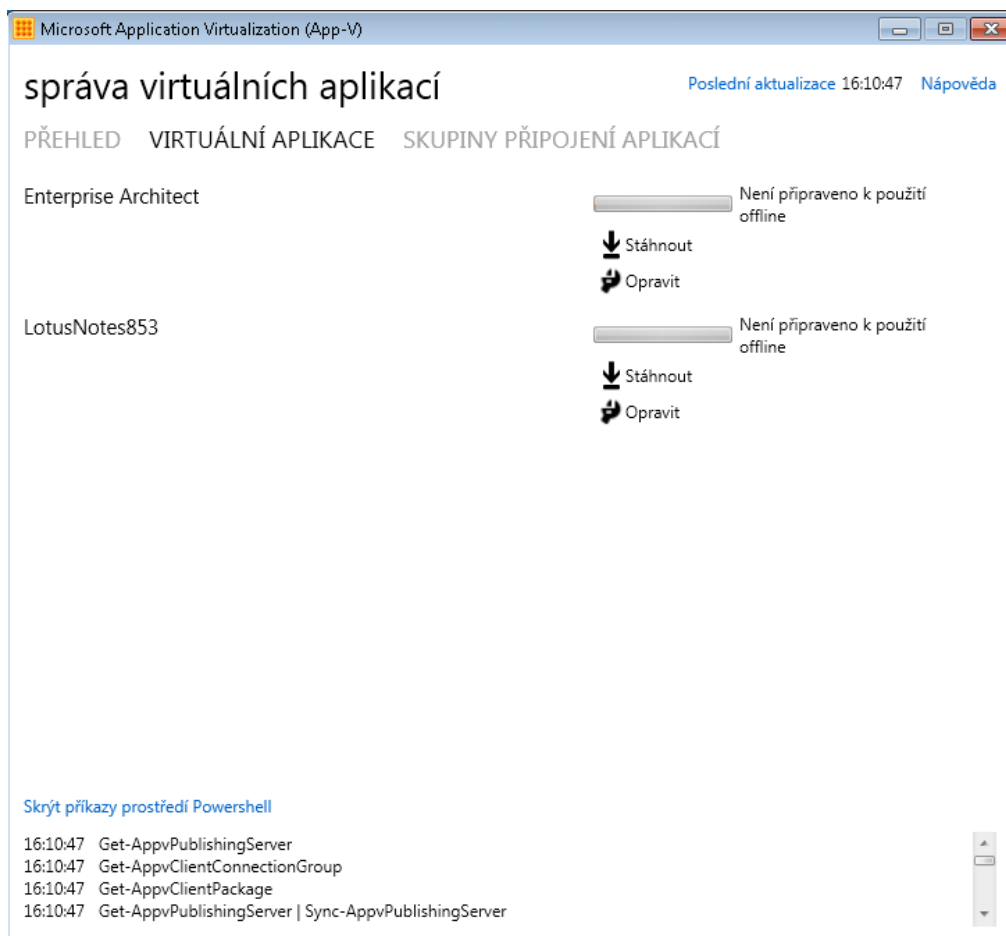
4.3.3 Práce s aplikací

Členění aplikace je rozděleno do třech základních sekcí. První záložkou je přehled, kde jsou možnosti aktualizovat, stáhnout všechny virtuální aplikace a pracovat offline. Možnost aktualizovat ověří seznam dostupných aplikací pro daného klienta. Dále provede kontrolu, zda nebyla publikována novější verze již stažené virtuální aplikace. Druhá možnost umožňuje stáhnout všechny dostupné virtuální aplikace do lokální mezipaměti pro použití mimo síť. Poslední volba přepne klienta do režimu offline, ve kterém nebudou přijímány žádné informace ze strany serveru. Uživatelské rozhraní klientské aplikace je zobrazeno na následujícím obrázku.



Obrázek 15 - Úvodní obrazovka klienta, zdroj: vlastní pořízení

Pod záložkou virtuální aplikace lze zobrazit seznam využívaných virtuálních aplikací. Kromě seznamu všech využívaných virtuálních aplikací je zobrazována informace o aktuálním stavu dané aplikace. Virtuální aplikace není po synchronizaci se serverem schopna pracovat v režimu offline. Aplikaci lze spustit a pracovat s ní, ovšem pouze za předpokladu připojení k síti. O této skutečnosti jsme informováni v záložce virtuální aplikace. Na následujícím obrázku je zobrazena virtuální aplikace Enterprise Architekt, která byla publikována před spuštěním klienta. Z obrázku je patrné, že aplikaci nelze používat v offline režimu.



Obrázek 16 - Virtuální aplikace v režimu online, zdroj: vlastní pořízení

V případě potřeby práce s virtuální aplikací mimo podnikovou síť lze aplikaci stáhnout a využívat v režimu offline. Stačí zvolit požadovanou aplikaci a kliknout na možnost stáhnout. Druhou možností je v přehledu na úvodní obrazovce zvolit možnost stáhnout všechny virtuální aplikace. Po přenesení veškerých dat jsme informováni o možnosti využití virtuální aplikace v režimu offline. V tento okamžik již nejsme limitováni využíváním aplikací pouze v případě, kdy jsme připojeni dopodnikové síť.

5 Zhodnocení výsledků a doporučení

Na základě analýzy technologií zabývajících se virtualizací aplikací byla pro praktické znázornění vybrána technologie App-V od společnosti Microsoft. Implementace probíhala ve třech krocích. Nejprve byla implementována komponenta Sequencer, která vytváří virtuální aplikační balíčky. Ta je provozována na virtuálním počítači z důvodu ukládání stavu virtuálního stroje. Ten je po vytvoření virtuální aplikace vrácen do původního stavu před instalací. Tím je dosaženo konzistentního prostředí. Pro demonstrování fungování aplikační virtualizace byl pomocí aplikace Sequencer vytvořen balíček s aplikací Enterprise Architect. Další implementovanou komponentou byl virtualizační server. Ten obstarává správu a distribuci virtuálních aplikací. Dříve vytvořený aplikační balíček byl importován na server a byla provedena jeho konfigurace. Posléze byl publikován na koncové stanice. Na koncových stanicích byl implementován klient zajišťující komunikaci se serverem a spouštění vizualizovaných aplikací.

V rámci testování zvolené technologie bylo vybráno celkem pět aplikací, které byly postupně implementovány. Z těchto pěti aplikací se podařilo úspěšně nasadit čtyři. V případě neúspěšně nasazené aplikace se jednalo o poštovního klienta, který byl v průběhu testování nahrazen novým řešením. Z tohoto důvodu nebylo v implementaci dále pokračováno. Celková úspěšnost nasazení aplikací je tedy 80%. V rámci testování dané technologie bylo od uživatelů zadáno celkem 15 incidentů. V nadpoloviční většině případů se jednalo o nedostupnost síťového připojení na koncových stanicích. Ve dvou případech byl zaznamenán problém, kdy aplikace po krátký časový úsek neodpovídala. Situace nastávala zejména při spuštění většího počtu aplikací. Dále byl zaznamenán případ, kdy se klientská aplikace samovolně ukončovala. Problém byl vyřešen reinstalací klientské aplikace.

Vzhledem ke zkušenostem s využíváním virtualizace aplikací lze technologii hodnotit jako přínosnou pro správu aplikačního portfolia. Centrální správa virtuálních aplikací umožňuje rychlou a cílenou distribuci aplikací na koncové zařízení. Oproti klasickému přístupu přináší výhodu v podobě izolací aplikací od operačního systému. To prodlužuje životnost operačního systému a eliminuje případné konflikty mezi aplikacemi.

S ohledem na průběh testování je doporučeno setrvat v testovacím režimu. Dále se zaměřit na rozšíření portfolia testovaných aplikací, aby byla získána komplexnější

představa o úspěšnosti nasazovaných aplikací. V neposlední řadě je potřeba se zaměřit na aktualizaci App-V. Od doby nasazení byl již vydán servisní balíček s pořadovým číslem tři. Ten vylepšuje dosavadní funkce a přináší větší stabilitu systému.

6 Závěr

Hlavním cílem byla identifikace aktuálních technologií v oblasti virtualizace aplikací a následná implementace zvolené technologie v reálném prostředí. Identifikace virtualizačních technologií byla provedena v rámci teoretické části, kde byl vytvořen základní přehled o jednotlivých přístupech k virtualizaci. V praktické části práce byl popsán postup implementace a používání technologie App-V. Dále byly realizovány i dílčí cíle práce:

- *Zdůraznit teoretické principy virtualizace*

Hlavní princip virtualizace aplikací spočívá v oddělení aplikace od operačního systému. Díky této vlastnosti se při používání virtuálních aplikací nemění stav pracovní stanice, což prodlužuje její životnost. Virtuální aplikace jsou navíc izolovány i samy od sebe, díky tomu jsou vyloučeny vzájemné konflikty.

Virtualizace dále umožňuje centralizovanou správu a distribuci virtuálních aplikací, což přispívá k zefektivnění správy aplikačního portfolia. Virtuální aplikace jsou zabezpečovány díky streamovacím a virtualizačním agentům, kteří umožňují specifikovat přístupová oprávnění.

- *Analyzovat a porovnat virtualizační technologie*

Zde je nutné zmínit, že jednotlivé technologie využívají podobných principů. V rámci analýzy byly identifikovány stěžejní vlastnosti, které umožňují porovnání konkrétních technologií. Mezi ně patří například izolace aplikací, existence agentů a možnost streamování aplikací. Tyto vlastnosti představovali kritéria pro Saatyho metodu. Tato metoda slouží k stanovení vah zvolených kritérií. Na základě této metody byla vybrána technologie App-V.

- *Implementovat vybrané řešení virtualizace aplikací*

V praktické části byla implementována vybraná technologie App-V. Implementace byla rozdělena do třech logických částí. Nejprve byla implementována aplikace Sequencer pro tvorbu virtuálních aplikací. Následovala serverová část obstarávající správu a distribuci aplikací. Poslední komponentou byla klientská část zajišťující stahování a spouštění virtuálních aplikací. Po implementační fázi následovalo testování. Testováno bylo celkem

pět aplikací, z nich se jedna nepodařila úspěšně nasadit. U ostatních aplikací lze konstatovat úspěšné nasazení a používání.

- *Formulovat obecné i specifické závěry*

Na základě získaných zkušeností s aplikační virtualizací lze konstatovat, že tato technologie představuje užitečný nástroj pro správu aplikací. Umožňuje rychlý a flexibilní způsob nasazení aplikací. Nicméně představuje poměrně značný zásah do stávající infrastruktury, jelikož stávající aplikace je nutné odinstalovat a nahradit virtuálními. Z tohoto důvodu je vhodné naplánovat přechod k virtualizaci v rámci plánované obměny koncových stanic.

Z širšího hlediska aplikační virtualizace přináší nezpochybnitelné přínosy v oblasti informačních technologií. Její využití lze aplikovat v různorodých organizacích, ať už se jedná o firmy, školy nebo státní správu. S rozvojem výpočetní techniky je čím dál tím více kladen důraz na snižování provozních nákladů ICT. Virtualizace aplikací je jednou z možností jak tohoto cíle dosáhnout, při zachování dostupnosti a kvality služeb. To je jeden z důvodů, proč lze očekávat další rozvoj aplikační virtualizace.

7 Seznam použitých zdrojů

Knihy:

1. RUEST, Danielle a Nelson RUEST. *Virtualizace: podrobný průvodce*. Vyd. 1. Překlad Pavel Vaida. Brno: Computer Press, 2010. ISBN 978-80-251-2676-9.
2. OLZAK, Thomas a Ken MAJORS. *Microsoft virtualization: master Microsoft server, desktop, application, and presentation virtualization*. Boston: Syngress/Elsevier, c2010. ISBN 1597494313.
3. RUEST, Danielle a Nelson RUEST. *Virtualization: a beginner's guide*. New York: McGraw Hill, c2009. Network professional's library. ISBN 007161401X.
4. KELBLEY, John a Mike STERLING. *Microsoft Windows Server 2008 R2 Hyper-V: podrobný průvodce administrátora*. Vyd. 1. Brno: Computer Press, 2011. ISBN 978-80-251-3286-9.
5. ALVAREZ, Augusto. *Getting started with Microsoft application virtualization 4.6: virtualize your application infrastructure efficiently using Microsoft App-V*. Olton, Birmingham: Packt Pub. Ltd., 2011.

Elektronické dokumenty:

6. Appv04. Augustoalvarez [online]. [cit. 2016-03-20]. Dostupné z: <https://augustoalvarez.files.wordpress.com/2009/07/appv04.png>
7. VMware ThinApp: Reviewers guide [online]. 2011, , 67 [cit. 2016-03-30]. Dostupné z: <http://www.vmware.com/files/pdf/VMware-ThinApp-Reviewers-Guide.pdf>
8. Use application virtualization before moving to VDI. Techtarget [online]. 2008 [cit. 2016-03-20]. Dostupné z: <http://searchvirtualdesktop.techtarget.com/tip/Use-application-virtualization-before-moving-to-VDI/>
9. VÝŠEK, Ondřej. Microsoft Application Virtualization (App-V) přehled. Optimalizovane-it [online]. 2009 [cit. 2016-03-20]. Dostupné z: <http://www.optimalizovane-it.cz/aplikacni-virtualizace/microsoft-application-virtualization-app-v-prehled.html>
10. Microsoft Desktop Optimization Pack. Technet [online]. [cit. 2016-03-20]. Dostupné z: <https://technet.microsoft.com/en-us/windows/bb899442>

11. *App-V: Microsoft Application Virtualization. Technet [online]. 2011 [cit. 2016-03-20]. Dostupné z: <http://social.technet.microsoft.com/wiki/contents/articles/2717.app-v-microsoft-application-virtualization.aspx>*
12. *So Then What is App-V? Technet [online]. [cit. 2016-03-20]. Dostupné z: <http://blogs.technet.com/b/canitpro/archive/2009/11/27/so-then-what-is-app-v.aspx>*
13. *VMWARE THINAPP. Oldanygroup [online]. [cit. 2016-03-20]. Dostupné z: <http://www.oldanygroup.cz/vmware-thinapp>*
14. *Metody stanovení vah kriterií. VSB [online]. [cit. 2016-03-20]. Dostupné z: <http://kds.vsb.cz/mhd/kvalita-vahy.htm>*
15. *Aplikační virtualizace a distribuce virtualizovaných aplikací pomocí System Center Configuration Manager 2007 R2 (1. díl). Technet [online]. 2010 [cit. 2016-03-20]. Dostupné z: <https://blogs.technet.microsoft.com/technetczsk/2010/06/29/aplikan-virtualizace-a-distribuce-virtualizovanych-aplikac-pomoc-system-center-configuration-manager-2007-r2-1-dl/>*